공학박사학위논문

Style Adaptive Speech Synthesis Using Neural Network Approaches

딥러닝을 활용한 스타일 적응형 음성 합성 기법

2020년 8월

서울대학교 대학원

전기 · 컴퓨터공학부

이 준 엽

Style adaptive speech synthesis using neural network approaches

딥러닝을 활용한 스타일 적응형 음성 합성 기법

지도교수 김 남 수

이 논문을 공학박사 학위논문으로 제출함

2020년 7월

서울대학교 대학원

전기 · 컴퓨터공학부

이 준 엽

이준엽의 공학박사 학위논문을 인준함

2020년 06월

위 원 장 김 성 철 (인)

부위원장 김 남 수 (인)

위 원 심 병 효 (인)

위 원 장 준 혁 (인)

위 원 신 종 원 (인)

Abstract

The neural network-based speech synthesis techniques have been developed over the years. Although neural speech synthesis has shown remarkable generated speech quality, there are still remaining problems such as modeling power in a neural statistical parametric speech synthesis system, style expressiveness, and robust attention model in the end-to-end speech synthesis system. In this thesis, novel alternatives are proposed to resolve these drawbacks of the conventional neural speech synthesis system.

In the first approach, we propose an adversarially trained variational recurrent neural network (AdVRNN), which applies a variational recurrent neural network (VRNN) to represent the variability of natural speech for acoustic modeling in neural statistical parametric speech synthesis. Also, we apply an adversarial learning scheme in training AdVRNN to overcome the oversmoothing problem. From the experimental results, we have found that the proposed AdVRNN based method outperforms the conventional RNN-based techniques.

In the second approach, we propose a novel style modeling method employing mutual information neural estimator (MINE) in a style-adaptive end-to-end speech synthesis system. MINE is applied to increase target-style information and suppress text information in style embedding by applying MINE loss term in the loss function.

The experimental results show that the MINE-based method has shown promising performance in both speech quality and style similarity for the global style token-Tacotron.

In the third approach, we propose a novel attention method called memory attention for end-to-end speech synthesis, which is inspired by the gating mechanism of long-short term memory (LSTM). Leveraging the gating technique's sequence modeling power in LSTM, memory attention obtains the stable alignment from the content-based and location-based features. We evaluate the memory attention and compare its performance with various conventional attention techniques in single speaker and emotional speech synthesis scenarios. From the results, we conclude that memory attention can generate speech with large variability robustly.

In the last approach, we propose selective multi-attention for style-adaptive end-to-end speech synthesis systems. The conventional single attention model may limit the expressivity representing numerous alignment paths depending on style. To achieve a variation in attention alignment, we propose using a multi-attention model with a selection network. The multi-attention plays a role in generating candidates for the target style, and the selection network choose the most proper attention among the multi-attention. The experimental results show that selective multi-attention outperforms the conventional single attention techniques in multi-speaker speech synthesis and emotional speech synthesis.

Keywords: Neural SPSS, AdVRNN, end-to-end speech synthesis, style-adaptive speech synthesis, MINE, memory attention, selective multi-attention.

Student number: 2013-20858

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Speech is one of the most widely used interfaces to communicate within the human-human relationship. Speech delivers not only text information but also expresses emotion and mood with diverse speaking styles. With the rapid growth of technologies these days, interfaces of human-machine interaction are diversifying from keyboard, mouse, touch-pad to speech. Especially, speech plays an essential role in communicating with human-machine interaction in lots of smart devices. To utilize speech as an interface, speech should be used in two ways, recognition and generation. For recognition and generation, speech recognition systems and speech synthesis systems have been researched for years. Speech synthesis system aims to generate human-like speech given corresponding text input. Speech synthesis system, generally known as text-to-speech synthesis (TTS), an arbitrary given text is transformed into a synthetic speech signal. To generate the speech signal closed to the human voice, it is required to have an ability to reproduce intelligible and natural-sounding

speech with arbitrary speaker's voice characteristics and speaking styles such as emotion, prosody, or singing voice. Speech synthesis technology is now widely used to communicate with smart devices such as smartphones, navigation systems. Also, a speech synthesis system is widely used to announce information to humans, such as automatic response service, notification system with speech, audiobook, etc.

Until the appearance of neural network technology, the most popular approach was the concatenative method and hidden Markov model (HMM)-based statistical parametric speech synthesis (SPSS) system. In the 1990s, the most popular approach for speech synthesis was a concatenative method where appropriate samples or units of speech are selected according to the corresponding text from a large speech database [1] [2]. It is also known as a unit selection, which has been shown to synthesize a high quality synthesized speech and acquired a high-quality reputation from people.

Even the concatenative method or unit selection generates high-quality samples, there are several restrictions to be in wide use. First, it requires a huge database to make a high-quality sample with a consistent speaking pattern. This means the speakers are limited to be skillful ones such as announcers or voice actors, which yield lots of costs to construct the database. Moreover, to make expressive speeches, the same number of databases is required to reconstruct similar reading-styled quality. Since unit selection approaches use small speech units, it has an inflexible model, which is hard to change its speech characteristic.

Since 2000 to 2013, HMM-based SPSS system is in popularity [3], [4]. In this system, context-dependent HMMs are trained from speech database, and synthesized speech waveform is generated from the HMMs chosen by the given sentence. Although the overall quality of HMM-based synthesized speech is relatively lower

than that of the unit selection-based systems, due to HMM-based SPSS is based on a parametric approach, it offers the ability to model diverse styles without requiring large database. The styles are not limited to the speaking styles, any of emotions, multi-speaker is possible to adapt the voice model using a small number of target voice database. SPSS systems typically have a separate linguistic module, acoustic model, duration model, vocoder. Thus, errors in each module can be accumulated and yield quality degradation.

In recent years, neural speech synthesis techniques have shown remarkable improvement in speech quality. The early approach of neural speech synthesis replaces the HMM-based acoustic model with neural network (e.g., feedforward network, DBLSTM [5], [6]) which has strengths in modeling the nonlinear relationship between input and output. More recently, end-to-end speech synthesis such as Tacotron [7], [8], Deep Voice [9]–[11], DCTTS [12], Transformer-TTS [13], VoiceLoop [14], Char2Wav [15], Fastspeech [16] take place in neural text-to-speech(TTS) field. The end-to-end speech synthesis system merges separate text analysis modules, duration module, and an acoustic model with a unified encoder-decoder model. The end-to-end speech synthesis techniques can be implemented with relatively simpler network architecture design while requiring less linguistic knowledge than the conventional methods with high performance.

1.2 Scope of thesis

This thesis proposes four kinds of algorithms to enhance the performance of the neural speech synthesis system.

For neural SPSS, we propose an acoustic modeling technique applying the adver-

sarially trained variational recurrent neural network (AdVRNN) as an alternative to the conventional RNNs. AdVRNN is a modified version of VRNN, which adjusts autoencoder to the encoder-decoder model as in [17]. The AdVRNN is capable of modeling variability in a sequence efficiently than the vanilla recurrent neural network model due to latent random variables.

Then, we propose a novel style modeling method employing mutual information (MI) for end-to-end speech synthesis. MI is applied to increase target-style information and suppress text information in style embedding by adding MI loss term in the objective function. To estimate MI for neural networks, we adopt mutual information neural estimator (MINE).

For style-adaptive end-to-end speech synthesis, we propose a novel attention algorithm called memory attention, which is inspired by a gating technique in long-short term memory (LSTM) [18]. LSTM can capture the long-term information of the feature sequence applying the memory gates. We apply a similar gating method to the attention mechanism so that the attention alignment path can also be controlled precisely according to both the location information (previous alignment) and content information (input sequence and output sequence). With these gates, we can alleviate attention failure (e.g., skipping, repeating, murmuring of phones).

Also, for style-adaptive end-to-end speech synthesis, we propose selective multi-attention (SMA) to capture the alignment variability depending on style. To improve the alignment variability representing the attention model's capacity over the conventional single attention, we use multiple attentions. Also, we adopt a selection network that learns to select an appropriate attention model for the target speaker within multiple attentions. As a result, we can achieve a diverse attention path with SMA.

The rest of the thesis is organized as follows: The next chapter introduces the fundamentals of the neural speech synthesis system. In Chapter 3, we present our proposed AdVRNN for neural SPSS. In Chapter 4, we propose MINE based training for style-adaptive end-to-end speech synthesis systems. In Chapter 5 and 6, we proposed advanced attention methods of style-adaptive end-to-end speech synthesis. The conclusions are drawn in Chapter 7.

Chapter 2

Neural Speech Synthesis System

This chapter gives a brief introduction to the neural speech synthesis systems. In the early approach in the neural speech synthesis system, neural statistical parametric speech synthesis (SPSS) systems are proposed. Neural SPSS has a separate text processing module, duration model, acoustic model, and vocoder. On the other hand, end-to-end speech synthesis integrates such separate modules into a unified neural network framework. End-to-end speech synthesis systems typically have shown relatively better synthesized speech quality than the neural SPSS systems. In this chapter, we describe the overall structure and the basic algorithms of the neural SPSS system and end-to-end speech synthesis system.

2.1 Overview of a Neural Statistical Parametric Speech Synthesis System

The neural SPSS systems are inspired by the conventional hidden Markov model (HMM)-based SPSS system. Typical neural SPSS systems replace HMM in the

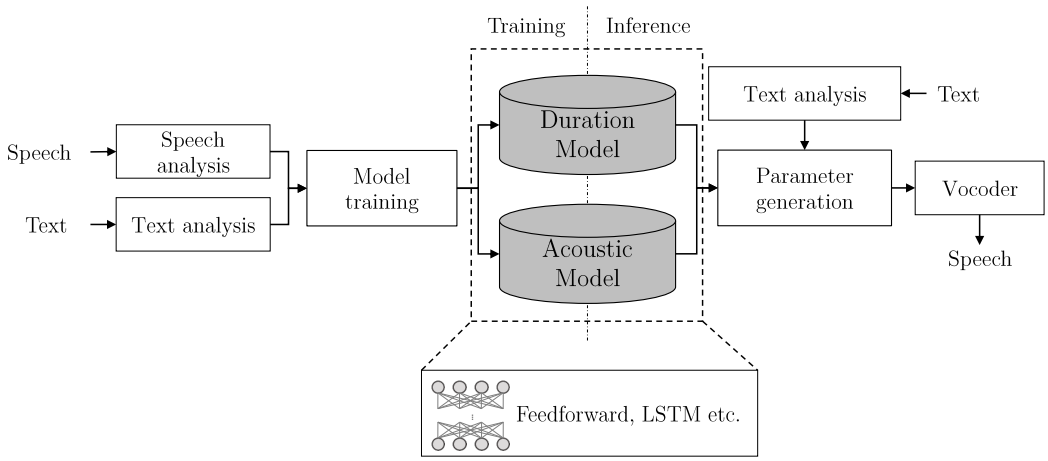


Figure 2.1: A block-diagram of the neural SPSS system.

acoustic model and duration model to the neural network mechanism. Figure 2.1 shows a block diagram of the neural SPSS system. The system consists of two stages: training and synthesis.

In the training stage, the acoustic model and duration model are trained using the given database. Spectrum and excitation are extracted from each frame of the speech signals, and their static and dynamic features are employed as observation features to estimate the parameters of the neural network. The models are context-dependent, and they consider phonetic, linguistic, and prosodic contexts. These contexts include local contexts (e.g., phoneme identity or adjacent phonemes) and global contexts analyzed from the whole sentence (e.g., stress-related contexts and locational information). Such context information is added at text analysis and expressed as one-hot or numerical input linguistic features. As speech is a sequential signal, recurrent neural network (RNN)-based acoustic model such as long-short term memory (LSTM)-based acoustic model [6] shows better performance than feedforward network-based approach [5]. Also, the duration model and acoustic model

are trained separately, and require a pre-aligned database or additional alignment process.

In the synthesis stage, at first, a sequence of context-dependent labels are interpreted by text analysis from a given input sentence. From the label sequence, the duration is determined using the duration model. Based on the duration, text analysis output is broadcasted according to duration, then input to the acoustic model. After that, spectral and excitation feature parameter sequences are obtained by the parameter generation. Finally, a speech signal is synthesized from the feature sequences by the vocoder.

As the SPSS system has a high ability to change the speech style, the neural SPSS also can be adapted to the target style with a small database. However, since neural SPSS uses separate modules, each module's errors can be accumulated, which yield performance degradation. Moreover, neural SPSS needs additional pre-alignment process of database and require linguistics knowledge to make context feature.

2.2 Overview of End-to-end Speech Synthesis System

Exploiting the success of neural sequence-to-sequence framework in machine translation [19] and speech recognition [20], [21], end-to-end speech synthesis systems such as Tacotron [7], [8], Deep Voice [9]–[11], DCTTS [12], Transformer-TTS [13], VoiceLoop [14], Char2Wav [15], Fastspeech [16] have been developed and successfully deployed in various applications. The end-to-end speech synthesis techniques can be implemented with relatively simpler network architecture design while requiring less linguistic knowledge than the conventional SPSS methods. Numerous end-to-end speech synthesis frameworks utilize attention mechanisms to predict alignment be-

tween an input text sequence and a speech feature sequence. Since the attention mechanism is jointly trained within the end-to-end framework, it does not require either additional training processes nor a pre-aligned database. Thus, the end-to-end speech synthesis framework has contributed to lower the hurdle for developing practical TTS systems and has improved the overall synthesized speech quality.

State-of-the-art of end-to-end speech synthesis models are the Tacotron-based models, which are attention-based and autoregressive models with a neural vocoder. Neural vocoders can lead to human-like speech with an end-to-end speech synthesis model. Neural vocoder such as Wavenet [22], Parallel Wavenet [23], Parallel WaveGAN [24], and WaveGLOW [25] are widely used in these days. However, it needs additional training process and slows down the speed at the inference stage. For this reason, in this thesis, we used Griffin-Lim vocoder [26] instead of a neural vocoder.

2.3 Tacotron2

In this section, we will give a brief review of Tacotron2 [8], which is one of the widely used end-to-end models in speech synthesis field. Typically, the structure of the attention-based end-to-end speech synthesis systems such as Tacotron, Tacotron2, and DCTTS [7], [8], [12] consist of an encoder and an attention decoder. The encoder, which consists of multi-layer convolution networks and a single layer of bidirectional LSTM, outputs a text embedding sequence $\mathbf{x} = [x_1, x_2, ..., x_N]$ given an input text sequence $\mathbf{l} = [l_1, l_2, ..., l_N]$. The attention algorithm based on LSA summarizes the text embedding sequence as a context vector for each decoder time-step t. Then, the decoder predicts the mel-spectrogram sequence autoregressively using this context vector. At the decoder, previously generated mel-spectrogram is passed

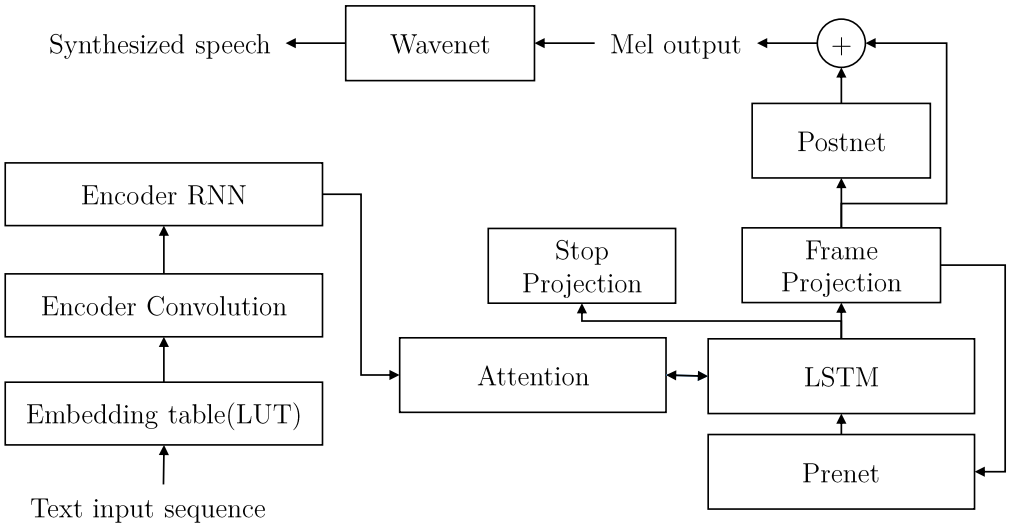


Figure 2.2: A block-diagram of the Tacotron2 system.

through a prenet, a multi-layer network with fully connected layers. The prenet output and the attention context vector are concatenated for decoder LSTM input. Then, the decoder LSTM output and the attention context vector are passed to a linear projection layer to predict the target mel-spectrogram and also are passed through another linear projection layer to predict the stop token. To enhance the reconstruction performance of the target mel-spectrogram, a postnet is used to predict the residual. The predicted mel-spectrogram sequence is passed to the WaveNet vocoder [22] to generate the speech waveform. However, in this thesis, instead of using the WaveNet, Griffin-Lim vocoder is used with the CBHG post-filter which converts the mel-spectrogram to the linear-spectrogram to reduce the training and inference time.

Let a decoder LSTM state sequence be $\mathbf{h} = [h_1, h_2, ..., h_T]$ and predicted mel-spectrogram $\hat{\mathbf{m}} = [\hat{m}_1, \hat{m}_2, ..., \hat{m}_T]$. In Tacotron2, $\hat{\mathbf{m}}$ is derived from $\mathbf{l}$ as follows:

$$\mathbf{x} = \text{Encoder}(\mathbf{l}), \tag{2.1}$$

$$h_t = \text{LSTM}_{\text{dec}}(h_{t-1}, c_{t-1}, \text{Prenet}(\hat{m}_{t-1})), \tag{2.2}$$

$$\alpha_t = \text{Atten}(\mathbf{x}, h_t), \quad c_t = \sum_{n=1}^{N} \alpha_{t,n} x_n, \tag{2.3}$$

$$\hat{m}_t^{'} = \text{LinearProj}(h_t, c_t), \quad \hat{m}_t = \text{Postnet}(\hat{m}_t^{'}) + \hat{m}_t^{'}, \tag{2.4}$$

where α_t, and c_t denote t-th element of attention alignment path and t-th context vector respectively. Finally, using neural vocoder(e.g. Wavenet [22]), predicted mel-spectrogram is converted to waveform.

2.4 Attention Mechanism

As aforementioned in Section 2.2, attention model makes alignment between text embedding sequence and speech feature sequence. There have been several attempts to make a suitable attention mechanism for speech synthesis such as Location sensitive attention (LSA), Forward attention (FA), dynamic convolution attention (DCA) etc. To achieve a human-like speech, the alignment path constructed by the attention mechanism in speech synthesis should stay or move forward for every decoder time-step monotonically.

2.4.1 Location Sensitive Attention

In the vanilla Tacotron2 system, LSA is used for the attention mechanism [8]. LSA combines content-based and location-based attention to mitigate the monotonic problem. From the content-based attention perspective, the energy function

of LSA is defined by the summation of the text embedding sequence term $\mathbf{V}x_n$ and the decoder LSTM state sequence term $\mathbf{W}h_t$, where $\mathbf{V}$ and $\mathbf{W}$ each denote weight matrices. Also, from the location-based attention prospect, a static convolutional filter $\mathbf{F}$ is applied to the previous attention alignment α_{t-1} to stabilize the attention alignment for LSA. The overall LSA formulation is given as follows:

$$e_{t,n} = v^T \tanh(\mathbf{W}h_t + \mathbf{V}x_n + \mathbf{U}f_{t,n} + b), \tag{2.5}$$

$$f_t = \mathbf{F} * \alpha_{t-1}, \tag{2.6}$$

$$\alpha_{t,n} = \exp(e_{t,n}) / \sum_{m=1}^{N} \exp(e_{t,m}) = \text{softmax}(e_{t,1 \leq n \leq N}), \tag{2.7}$$

$$c_t = \sum_{n=1}^{N} \alpha_{t,n} x_n, \tag{2.8}$$

where $e_{t,n}$ and f_t denote the energy function and the convolutional feature, respectively, and $\mathbf{U}$ represents a weight matrix for the convolutional feature.

Although LSA is known to yield a more stable alignment path than the fully content-based attention, there still exists a variety of alignment failures at the inference stage.

2.4.2 Forward Attention

FA [27] is proposed to achieve a better monotonic alignment between the text embedding sequence and the mel-spectrogram sequence than LSA. At each decoder time-step, the alignment paths that satisfy the monotonic condition are computed recursively using the forward algorithm similar to the connectionist temporal classification model [28]. Also, a transition agent u_t is applied to help the FA to decide

whether to proceed or stay at each decoder time-step. The overall procedure of the FA is given as follows:

$$\alpha_t = \mathrm{LSA}(\mathbf{x}, h_t) \tag{2.9}$$

$$y'_t(n) = \{(1 - u_{t-1})y_{t-1}(n) + u_{t-1}y_{t-1}(n-1)\}\alpha_{t,n}, \tag{2.10}$$

$$y_t(n) = y'_t(n) / \sum_{m=1}^{N} \exp(y'_t(m)), \tag{2.11}$$

$$c_t = \sum_{n=1}^{N} y_t(n)x_n, \tag{2.12}$$

$$u_t = \mathrm{DNN}(c_t, m_{t-1}, h_t), \tag{2.13}$$

where y' and y denote the forward variable and normalized forward variable respectively, and DNN represents a single layer feedforward network with a sigmoid output activation. From the product-of-experts model perspective, forward attention can be seen as a product of the monotonic alignment constraint and attention alignments, which leads FA to have a low probability of violating the monotonic condition of attention. In consequence, FA results in better monotonic attention than the LSA, reducing the repeated words of synthesized speech.

2.4.3 Dynamic Convolution Attention

DCA [29] is a solely location-relative attention mechanism to generate long sentence without using content-based terms, i.e., $\mathbf{W}h_t$ and $\mathbf{V}x_n$ in the energy function. DCA only considers the previous alignment path α_{t-1} to determine the attention alignment at time t. To alleviate the problem of the static filters mentioned in Sec-

tion 2.4.1 and Section 2.4.2, DCA applies a dynamic filter $\mathbf{G}(h_t)$ which adjusts the alignment dynamically depending on the decoder LSTM state. Also, to prevent the attention from moving backward, DCA applies a fixed causal prior filter $\mathbf{P}$. The overall procedure of the DCA is given as follows:

$$e_{t,n} = v^T \tanh(\mathbf{U} f_{t,n} + \mathbf{T} g_{t,n} + b) + p_{t,n}, \tag{2.14}$$

$$g_t = \mathbf{G}(h_t) * \alpha_{t-1}, \quad \mathbf{G}(h_t) = \mathbf{V}_G \tanh(\mathbf{W}_G h_t + b_G), \tag{2.15}$$

$$\alpha_{t,n} = \text{softmax}(e_{t,n}), \tag{2.16}$$

$$\mathbf{p}_t = \log(\mathbf{P} * \alpha_{t-1}), \tag{2.17}$$

$$c_t = \sum_{n=1}^{N} \alpha_{t,n} x_n, \tag{2.18}$$

where g_t and $p_{t,n}$ denote the dynamic convolutional feature and fixed prior filter-based bias, respectively. The dynamic filter $\mathbf{G}$ is learned as in Equation (2.15) where $\mathbf{V}_G$, $\mathbf{W}_G$ and b_G denote weight matrices and bias for dynamic filter.

DCA shows a better performance especially in generating long sentences compared to the LSA method while preserving naturalness for shorter in-domain sentences.

Chapter 3

Neural Statistical Parametric Speech Synthesis using AdVRNN

3.1 Introduction

Since speech has complex time-dependencies, recurrent neural networks (RNNs) such as long short term memory (LSTM) [30]–[32], simplified LSTM [33], and gated recurrent unit (GRU) [17], [33], [34] have been applied to improve the performance in acoustic modelling for neural statistical parametric speech syntheis (SPSS).

Although standard RNNs introduced in [17], [30], [31] have improved the performance over the non-recurrent deep neural networks, such methods have difficulty in capturing the variability in data due to the entirely deterministic structures. Variational recurrent neural network (VRNN) is introduced in [35] to model the variability in highly structured sequential data such as natural speech or handwrit-

ing. The VRNN generates an estimated input-like sequence conditioned on latent random prior and RNN state variables.

In this chapter, we propose an acoustic modeling technique using the adversarially trained variational recurrent neural network (AdVRNN) as an alternative to the conventional RNNs for SPSS which is inspired by VRNN. Unlike the VRNN, the proposed AdVRNN for SPSS takes the input of a linguistic feature sequence with the latent random prior and the RNN state variable. AdVRNN for SPSS generates not a sequence of linguistic features but a sequence of acoustic features. In this regard, AdVRNN is closer to the encoder-decoder model in [17] than an autoencoder. The AdVRNN is capable of modeling variability in a sequence efficiently than the vanilla encoder-decoder model due to latent random variable.

For training AdVRNN, an adversarial training scheme similar to the generative adversarial networks [36] is employed to capture the detailed structure of real acoustic features. A discriminator is introduced during the training phase to distinguish the generated acoustic feature sequence from the real data sequence. To avoid oversmoothing, which is one of the significant problems in speech synthesis, a method to reduce the cost of discriminator was used instead of maximizing the variational lower bound. Unlike the sampled noise prior in [36], a latent random variable inferred from linguistic features and the RNN state variables are used to generate acoustic features in AdVRNN. It is shown that the proposed method performs better than conventional RNN based speech synthesis such as GRU for both objective and subjective measures. The detailed structure and training scheme of the AdVRNN for SPSS are introduced in Section 3.3.

3.2 Background

In this section, we will give a brief review of the conventional VAE and VRNN.

3.2.1 Variational Autoencoder

By employing the structure of autoencoder, where the network aims to generate the input at the output layer, VAE introduces latent random variable to apply stochastic component in autoencoder and model the variations in observations [37]. VAE is composed of an encoder network which maps the observed data $\mathbf{x}$ (e.g., acoustic parameters in speech application) to latent random variable $\mathbf{z}$, and a decoder network which maps the latent random variable $\mathbf{z}$ to the output $\mathbf{x}$ same as the input. The prior distribution of latent random variable $\mathbf{z}$ is usually assumed to be standard Gaussian. However, the difficulty in VAE comes from the intractability in inferencing the posterior distribution $p(\mathbf{z}|\mathbf{x})$. To overcome this challenge, VAE employs the variational approximated posterior $q(\mathbf{z}|\mathbf{x})$ with a neural network. Using $q(\mathbf{z}|\mathbf{x})$, the variational lower bound is derived as follows:

$$\log p(\mathbf{x}) \geq -D_{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})], \tag{3.1}$$

where D_{KL} refers to Kullback-Leibler divergence (KL divergence) between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$, measuring the similarity between the two distributions. Such KL divergence in theEquation 3.1 implies the regularization of the encoder parameters, and the remaining term refers to the reconstruction error between input and output of VAE. Both encoder and decoder are jointly trained by maximizing Equation 3.1.

3.2.2 Variational Recurrent Neural Network

With the highly structured sequential data, VAE can be extended into a recurrent neural network framework. This idea, known as the VRNN, can model highly non-linear dynamics of the sequential data and can capture the time dependency of the sequence [35]. Similar to VAE, VRNN is composed of an encoder and a decoder.

Decoder

Unlike the VAE, the prior of the latent random variable $\mathbf{z}_t$ of the VRNN is not standard Gaussian distribution but conditioned on RNN state variable $\mathbf{h}_{t-1}$ as follows:

$$p(\mathbf{z}_t) = \mathcal{N}(\mu_{p,t}, \sigma_{p,t}^2), \quad [\mu_{p,t}, \sigma_{p_t}] = \varphi^{pri}(\mathbf{h}_{t-1}), \tag{3.2}$$

where $\mu_{p,t}, \sigma_{p,t}$ denote the mean and standard deviation of the prior distribution, and φ^{pri} denotes neural network which models the prior distribution. The latent random variable can capture time dependency context using $\mathbf{h}_{t-1}$ in the prior distribution. The distribution of the generation model $p(\mathbf{x}_t|\mathbf{z}_t)$ is conditioned on $\mathbf{z}_t$ and $\mathbf{h}_{t-1}$ as follows:

$$p(\mathbf{x}_t|\mathbf{z}_t) = \mathcal{N}(\mu_{x,t}, \sigma_{x,t}^2), \quad [\mu_{x,t}, \sigma_{x,t}] = \varphi^{dec}(\phi^z(\mathbf{z}_t), \mathbf{h}_{t-1}), \tag{3.3}$$

where $\mu_{x,t}, \sigma_{x,t}$ denote the mean and standard deviation of generation model distribution and φ^{dec} is a deep neural network which captures the generation model distribution and ϕ^z is an embedding network of $\mathbf{z}_t$. RNN state variable $\mathbf{h}_t$, uses previous state variable $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$, $\mathbf{z}_t$ for updating the state variable as follows:

$$\mathbf{h}_t = \varphi^{rec}(\phi^x(\mathbf{x}_t), \phi^z(\mathbf{z}_t), \mathbf{h}_{t-1}), \tag{3.4}$$

where φ^{rec} is a state transition function in the RNN and ϕ^x is embedding networks of $\mathbf{x}_t$.

Encoder

The inference model for the encoder network at the **t**-th time-step can be expressed using the approximation of variational posterior $q(\mathbf{z}_t|\mathbf{x}_t)$ which is a function of $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$ as follows:

$$q(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mu_{z,t}, \sigma_{z,t}^2), \quad [\mu_{z,t}, \sigma_{z,t}] = \varphi^{enc}(\phi^x(\mathbf{x}_t), \mathbf{h}_{t-1}), \tag{3.5}$$

where $\mu_{z,t}, \sigma_{z,t}$ denote the mean and standard deviation of $q(\mathbf{z}_t|\mathbf{x}_t)$ and φ^{enc} is the deep neural network which captures approximated posterior distribution.

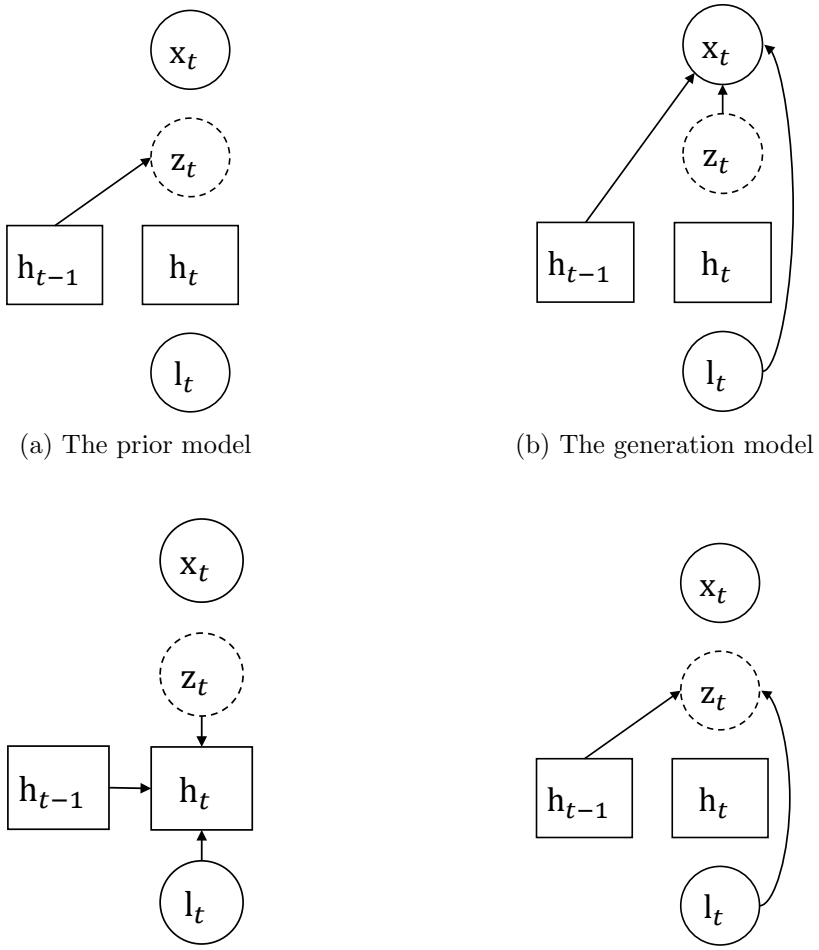
Variational Lower Bound

For learning, the variational lower bound in Equation 3.1 is modified as follows due to the time dependency:

$$\mathbb{E}_{q(\mathbf{z}_{\leq t}|\mathbf{x}_{\leq t})}\left[\sum_{t=1}^{T}(-D_{KL}(q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})\|p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{<t})) + \log p(\mathbf{x}_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}))\right]. \tag{3.6}$$

The concept of variational lower bound of VRNN is the same as VAE which consists of KL divergence term which regularizes the encoder parameters and reconstruction error term.

For more details about VAE and VRNN such as reparameterization tricks, the reader is referred to [35], [37].



(a) The prior model

(b) The generation model

(c) State transition model

(d) Inference model

Figure 3.1: Graphical representation for AdVRNN opearation.

3.3 Speech Synthesis Using AdVRNN

Using the idea of VRNN, we propose an AdVRNN-based speech synthesis technique which can model the complex nonlinear relation between linguistic feature sequence and acoustic feature sequence effectively. In this section, the acoustic model structure and the training procedure of the proposed technique are described.

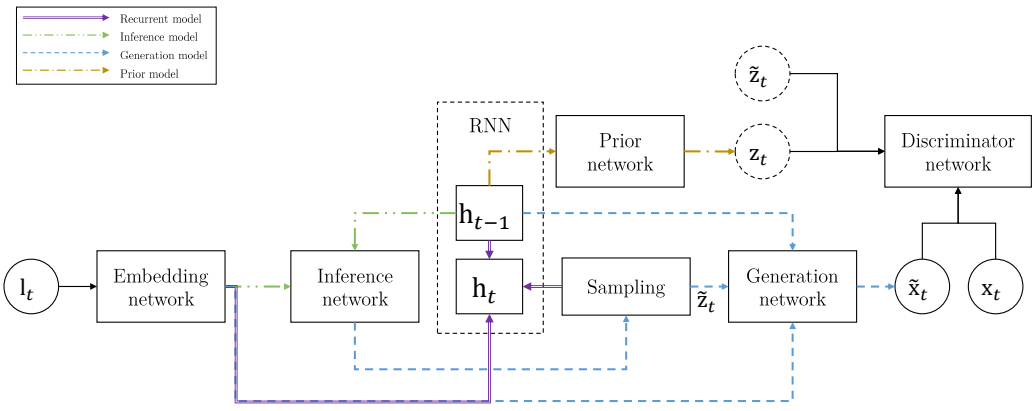


Figure 3.2: Overall training procedure of AdVRNN.

3.3.1 AdVRNN based Acoustic Modeling

Using the VRNN, as mentioned in section 3.2, observations can be generated using a latent random variable. Since a typical acoustic model in speech synthesis system takes a linguistic feature sequence as input and generates an acoustic feature sequence as output, the VRNN formulation is modified for speech synthesis application in a way shown in Figure 3.1.

Decoder

The prior distribution follows the same as in VRNN (i.e., Equation 3.2). However, due to the mapping between the linguistic feature sequence and the acoustic feature sequence in speech synthesis, the generation model distribution is conditioned not only on the latent variable $\mathbf{z}_t$ and the state variable $\mathbf{h}_{t-1}$, but also on the linguistic feature $\mathbf{l}_t$ as follows:

$$p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{l}_t) = \mathcal{N}(\mu_{x,t}, \sigma^2_{x,t}), \quad [\mu_{x,t}, \sigma_{x,t}] = \varphi^{dec}(\phi^z(\mathbf{z}_t), \phi^l(\mathbf{l}_t), \mathbf{h}_{t-1}), \tag{3.7}$$

where ϕ^l is the text embedding network. Applying the similar modification, the state update equation in the AdVRNN can be expressed as follows:

$$\mathbf{h}_t = \varphi^{rec}(\phi^l(\mathbf{l}_t), \phi^z(\mathbf{z}_t), \mathbf{h}_{t-1}). \tag{3.8}$$

Note that the above state update is missing $\mathbf{x}_t$ since in synthesis stage, the error in synthesized speech can propagate throughout the timesteps and amplify the error to cause performance degradation.

Encoder

The true posterior can be approximated with $q(\mathbf{z}_t|\mathbf{l}_t)$ conditioned on $\mathbf{l}_t$ and $\mathbf{h}_{t-1}$ as follows:

$$q(\mathbf{z}_t|\mathbf{l}_t) = \mathcal{N}(\mu_{z,t}, \sigma^2_{z,t}), \quad [\mu_{z,t}, \sigma_{z,t}] = \varphi^{enc}(\phi^l(\mathbf{l}_t), \mathbf{h}_{t-1}). \tag{3.9}$$

3.3.2 Training Procedure

The variational lower bound for AdVRNN can be derived using a similar approach to Equation 3.6. However, the variational inference method is known to blur the generated samples in the image processing research area [36], [38], [39]. In the speech synthesis, this can cause oversmoothing of the generated speech which makes muffled sound. In this chapter, we propose to use an adversarial schemes to overcome the oversmoothing problems as in Figure 3.2. In Figure 3.2, brown, blue, purple, green lines indicate the prior model, generation model, recurrent model, and inference model respectively as mentioned in section 3.3.1.

Following a typical generative adversarial network, the proposed method uses a discriminator to distinguish the followings:

1. For a real data sequence $\mathbf{x}_{\leq t}$, generate the corresponding latent variable sequence $\tilde{\mathbf{z}}_{\leq t}$ using inference model as defined in Equation 3.9

2. Generate $\tilde{\mathbf{x}}_{\leq t}$ and $\mathbf{z}_{\leq t}$ from the generation model and the prior model as defined in Equation 3.2 and 3.7, respectively.

Then, a discriminator is built to discriminate between $\{\mathbf{x}_t, \tilde{\mathbf{z}}_t\}$ and $\{\tilde{\mathbf{x}}_t, \mathbf{z}_t\}$.

Discriminating $\{\mathbf{z}_t, \tilde{\mathbf{z}}_t\}$ is similar to the KL divergence term in Equation 3.6 and discriminating $\{\mathbf{x}_t, \tilde{\mathbf{x}}_t\}$ is similar with the reconstruction term in Equation 3.6, where it has similar meaning with variational lower bound. The rest of the training procedure follows the same as a typical GAN training [36].

3.4 Experiments

To evaluate the performance of the proposed AdVRNN-based speech synthesis system, we conducted some objective measurements and subjective listening tests were conducted.

For the experiments, we used an Korean speech database spoken by a professional male voice-actor. Speaker provided 2,250 utterances of narrative speech data amounting to about 230 minutes. Among 2,250 utterances, we used 2,000 utterances for training, 200 utterances for validation, and 50 utterances for the test. Each utterance was sampled at 16kHz and 20 ms Hamming window was applied with 5 ms frame shift for acoustic feature extraction. STRAIGHT vocoder was used to extract the acoustic feature [40]. For the spectrum feature, the 25th-order mel-scaled cepstrum vector was used, and for the excitation feature, 1-dimensional logarithmic fundamental frequency (lf0) and 5-dimensional band aperiodicity (bap) were used. To make a continuous lf0 sequence, the lf0 values of the unvoiced region were filled

during the normalization process. Also, dynamic feature Δ and $\Delta\Delta$ were attached for each feature. The extracted acoustic feature was normalized to follow white Gaussian in order to use the acoustic feature as the target $\mathbf{x}_t$ for the speech synthesis systems. For input linguistic feature, a 547-dimensional binary feature for categorical linguistic contexts and 12-dimensional numeric feature for numerical linguistic contexts, position and duration were used together.

We used a deep GRU-based deterministic system to compare with a small AdVRNN model, and deep bidirectional LSTM (DBLSTM) to compare with a large AdVRNN model. For simplicity, we will call deep GRU as GRU. GRU system was configured to have two GRU layers and two feedforward hidden layers with a rectified linear unit (ReLU). Every layer was consisted of 256 nodes. Also, for DBLSTM, one tanh feedforward layer and two DBLSTM is used with 512 nodes for each direction. For training, the GRU and DBLSTM-based speech synthesis system, the Adam optimizer in [41] was used.

For AdVRNN the configurations of model is as follows:

- ϕ^l, ϕ^z: two hidden layers with 256 ReLU nodes.
- ϕ^x: two hidden layers with 256 ReLU nodes for a small model and 512 ReLU nodes for a large model.
- φ^{pri}: two feedforward hidden layers with 256 ReLU nodes for a small model and 512 ReLU nodes for a large model. The output layer was composed of the linear layer for $\mu_{p,t}$ and the softplus layer for $\sigma_{p,t}$. The dimension of the latent random variable $\mathbf{z}_t$ was 256.
- φ^{dec}: two feedforward hidden layers for a small model and three layers for a

large model with 1024 tanh nodes. The output layer was composed of linear layer for $\mu_{x,t}$ and softplus layer for $\sigma_{x,t}$.

- φ^{enc}: two feedforward hidden layers with 512 ReLU nodes. The output layer was composed of linear layer for $\mu_{z,t}$ and softplus layer for $\sigma_{z,t}$.

- For RNN, we use a GRU with 32 nodes for a small model and 512 nodes for a large model.

- For discriminator, four feedforward hidden layers with the bottom two feedforward layers composed of ReLU layers with 256 nodes for $\mathbf{x}$ and 128 nodes for $\mathbf{z}$. Then, one feedforward ReLU layers with 256 nodes and feedforward hidden layer of a ReLU layer with 128 nodes were used. Finally, the output layer was 1 -dimensional sigmoid layer to output whether the inputs was real or not.

We used a Adagrad optimizer in [42] to train AdVRNN and we used Tensorflow [43], a library for deep learning, for both GRU, DBLSTM and AdVRNN implementations in our experiments.

Table 3.1: *Objective measurement of GRU and small AdVRNN model.*

	MCD	RMSE of f0	bap distance	STOI	PESQ
GRU	6.203	**23.061**	2.420	0.711	1.539
AdVRNN	**5.808**	24.386	**2.312**	**0.794**	**1.872**

Table 3.2: *Objective measurement of DBLSTM and large AdVRNN model.*

	MCD	RMSE of f0	bap distance	STOI	PESQ
DBLSTM	5.695	**12.009**	2.145	0.827	2.008
AdVRNN	**5.572**	13.145	**2.058**	**0.920**	**3.070**

3.4.1 Objective performance evaluation

For objective measure, we used the averaged mel-cepstral distance (MCD) in dB scale, root mean square error (RMSE) of f0 in Hz, and bap distance in dB scale. Also, short-time objective intelligibility (STOI) and perceptual evaluation of speech quality (PESQ) was utilized to measure the intelligibility and the perceptual performance of synthesized speeches. PESQ is designed for speech assessment of the narrow-band telephone networks and speech codecs. It is also widely used in the speech enhancement fields to measure the perceptual quality of the degraded speech. STOI is a state-of-the-art speech intelligibility estimator, which relies on the linear correlation of speech temporal envelopes. STOI and PESQ are estimated using ground truth alignment. The results of the objective performance tests are shown in Table 3.1 and Table 3.2.

The results show that the AdVRNN approach is more effective for modeling highly structured speech data than GRU and DBLSTM approach. However, the performance of GRU and DBLSTM is better in RMSE of f0 measurement. We consider the reason for f0 degradation is due to the unvoiced part of the normalization process. The interpolated part of the unvoiced region to make continuous lf0 does not account for a real lf0 value, and this can influence the lf0 decoding process. Therefore, the performance of the proposed system related to the f0 could be improved if an accurate continuous pitch contour is available. STOI results show that AdVRNN have better intelligibility than the conventional RNN approaches. Also from the PESQ results, the perceptual quality of AdVRNN outperforms the others.

Table 3.3: *Results of MOS test: GRU and small AdVRNN model.*

	GRU	AdVRNN
MOS	2.836 ± 0.102	3.623 ± 0.103

Table 3.4: *Results of MOS test: GRU and small AdVRNN model.*

	GRU	AdVRNN
MOS	3.914 ± 0.093	4.132 ± 0.083

3.4.2 Subjective performance evaluation

We also performed a subjective listening test to compare the AdVRNN with the GRU and DBLSTM-based speech synthesis. Eleven participants listened to 20 sentences from each method, in which the sentences were randomly chosen from 50 test sentences. Each listener was provided with the speech samples in random order and was asked to measure the speech quality in terms of the mean opinion score (MOS). Each subject provided scores in the range of [1,5] with a large value indicating high performance. The results are shown in Table 3.3 and Table 3.4. From the results, we can find that the proposed method outperformed the conventional RNN methods as PESQ scores. These results showed that the AdVRNN has better intelligibility and quality than the RNNs-based models.

3.5 Summary

In this chapter, we have proposed using a VRNN as an alternative method for acoustic modeling in speech synthesis system. Since speech contains high variability information, we applied the VRNN, which can efficiently express the variability within the highly structured data. Instead of using the conventional variational lower

bound, we used an adversarial training scheme to increase the dynamic range for synthesized speech data. We called this VRNN with an adversarial training scheme as AdVRNN. The experimental results showed that the proposed AdVRNN-based method outperformed the conventional RNN-based methods for acoustic modeling.

Chapter 4

Speech Style Modeling Method using Mutual Information for End-to-End Speech Synthesis

4.1 Introduction

In recent years, neural speech synthesis using end-to-end frameworks have shown remarkable speech quality for a single speaker with monotonous speech. However, for the style-adaptive speech synthesis (such as multi-speaker, emotion, etc.), it shows poor performance compared to the single speaker end-to-end speech synthesis with narrative speech data.

There have been several attempts to use the target style embedding vector with supervised training for style-adaptive speech synthesis. Style embedding such as look-up table or d-vector is used for these methods. However, these style embedding techniques do not guarantee the naturalness of style-adaptive speech since such style

embedding vector has limited information. In the look-up table method, it is hard to add a new style or synthesize speech for unseen style. Moreover, since d-vector is developed for speaker recognition, it has a strong ability to represent speaker identity but it eliminates detailed speech characteristic which is needed for speech synthesis.

Meanwhile, instead of using the target style embedding vector, there have been several attempts to apply the style of reference speech in end-to-end speech synthesis. In [44], reference encoder compresses reference speech's prosody and reflects its prosody to the speech synthesis system. Also, the global style token (GST) [45] extracts speech style and replicates the speaking style of the reference audio clip using style token and its weights. This GST method shows high performance to synthesize reference-like synthesized speech. However, since these techniques are using unsupervised training schemes, it is nearly impossible to obtain a specific target style (e.g., speaker).

In this chapter, we propose using mutual information neural estimator (MINE) to GST-based style-adaptive end-to-end speech synthesis to model target-style more intensely. To reflect target style (e.g., speaker identity) more specifically, we estimate mutual information with MINE to obtain high dependency between the style token layer output and the style embedding vector (e.g., d-vector). It is shown that the proposed method outperforms the conventional GST-Tacotron for the subjective measures. The detailed structure and training scheme of our proposed method is introduced in Section 4.3.

4.2 Background

In this section, we will give brief review of mutual information, MINE, and global style token end-to-end model to explain our model.

4.2.1 Mutual Information

Mutual information is a measure of the mutual dependency between the two random variables using entropy, which is one of the widely used measure in probability theory and information theory. Let $(\mathbf{X}, \mathbf{Y})$ are random variables then, mutual information $\mathbf{I}$ between these variables is expressed as follows:

$$\mathbf{I}(\mathbf{X}, \mathbf{Y}) := \mathbf{H}(\mathbf{X}) - \mathbf{H}(\mathbf{X} \mid \mathbf{Y}), \tag{4.1}$$

where $\mathbf{H}$ is entropy. The larger the mutual information, the stronger the dependency between $\mathbf{X}$ and $\mathbf{Y}$. Also, if $\mathbf{X}$ and $\mathbf{Y}$ are independent, mutual information will be 0. The mutual information can be expressed using Kullback-Leibler divergence (KLD) between the product of marginal distributions and the joint distribution of random variables as follows:

$$\mathbf{I}(\mathbf{X}, \mathbf{Y}) := \mathbf{D}_{KL}(\mathbf{P}_{XY} \| \mathbf{P}_X \otimes \mathbf{P}_Y), \tag{4.2}$$

where $\mathbf{D}_{KL}$ denotes KLD. Using KLD forms of the mutual information and the Donsker-Varadhan representation of the KLD, mutual information has a lower bound as follows:

$$\mathbf{I}(\mathbf{X}, \mathbf{Y}) \geq \mathbb{E}_{\mathbf{P}_{XY}}[\mathbf{T}_\omega] - \log \mathbb{E}_{\mathbf{P}_X \otimes \mathbf{P}_Y}[e^{\mathbf{T}_\omega}], \tag{4.3}$$

where $\mathbf{T}$ is an arbitrary function which satisfy integrability constraints.

4.2.2 Mutual Information Neural Estimator

Mutual information is difficult to calculate in case of the continuous random variables. Thus, although mutual information can be used as a loss function, it has been barely used for the neural networks. However, MINE solves this problem using the lower bound in Equation 4.3 to estimate continuous mutual information. Let $\mathbf{T}_\omega$ be the function modeled by a neural network which called statistics network, MINE is defined as follows:

$$\hat{I}_\omega(\mathbf{X}, \mathbf{Y})_n = \sup \mathbb{E}_{\mathbf{P}_{XY}}[\mathbf{T}_\omega] - \log \mathbb{E}_{\mathbf{P}_X \otimes \mathbf{P}_Y}[e^{\mathbf{T}_\omega}]. \tag{4.4}$$

For training, MINE repeats to sample from joint distribution and marginal distribution, then update the statistics network using back-propagation. MINE has strong consistency and can be used to train generative models such as generative adversarial network. For more details about mutual information and MINE the reader is referred to [46].

4.2.3 Global Style Token

Global style token (GST)-based Tacotron system is proposed to model style in speech database with unsupervised training. Utilizing trained style token, it is possible to control style in synthesized speech. The overall structure of the GST Tacotron system is shown in Figure 4.1. During the training, reference speech is passed to the reference encoder, then its output is passed to self-attention and style token layer sequentially. This style embedding is concatenated with the text embedding to be conditioned on the Tacotron model. Style is controlled with the weight parameter of each token, and it is randomly initialized for training. Also, to train the target speech's speaking

style, it uses the same speech sample for both reference speech and target speech.

There are two ways for the inference. One is to use reference audio directly to mimic reference audio speaking style and the other is controlling style with style token weight parameters. Since style token is determined within the training procedure, it needs an additional operation to estimate the style of each style token. As GST Tacotron is hard to know which style token corresponds to a specific style, it has some difficulty to predict the target style.

4.3 Style Token end-to-end speech synthesis using MINE

In this section, we propose utilizing MINE for GST training. To model the target style more specific to the GST-Tacotron model, it is useful to have that corresponding target-style embedding vector. Depending on the target style, style can be speaker characteristic, emotion, gender and so on. In this chapter, we assume the target style as speaker characteristic and use d-vector as speaker embedding vector. D-vector is one of the most widely used speaker embedding vector for speaker verification. D-vector is a fixed-length embedding vector that is trained with the generalized end-to-end loss for the speaker verification network.

The overall structure of our proposed model using MINE is shown in Figure 4.2. We utilize MINE to maximize the mutual information between the style token layer output and the target style embedding vector (e.g., d-vector) to apply the target style (e.g., speaker characteristic). For this reason, we added MINE loss term in the conventional GST Tacotron loss function as follows:

$$\text{Loss} = \text{Loss}_{taco} - \alpha \mathbf{I}_{MINE}(\mathbf{S}, \mathbf{S}_v), \tag{4.5}$$

where Loss, Loss_{taco}, and $\mathbf{I}_{MINE}$ respectively denote loss function for proposed method, loss function in vanilla GST-Tacotron, and MINE loss between style token layer output $\mathbf{S}$ and target style embedding vector $\mathbf{S}_v$. Also, α is used to determine how much MINE loss will be applied to the loss function, and it has value between 0 to 1. As mentioned in Section 4.2.1, MI is a measure of presenting the dependency between two random variables. Therefore, using loss function as Equation 4.5, the dependency between style token layer output and target style embedding vector will be increased. As a result, it is possible to adapt target-style efficiently in GST-Tacotron. As the conventional GST-Tacotron has difficulty to adapt target-style specifically, utilizing our MINE based method, we can alleviate such difficulty.

Moreover, while maximizing target-style embedding vector dependency, other information such as text information will be disentangled with style token layer output. Disentanglement of such information is a crucial point for the controllability and the performance of style-adaptive speech synthesis. The inference procedure is the same as 4.2.3, since MINE network is used only for the training.

4.4 Experiments

We compared the proposed method with the conventional GST-Tacotron to evaluate the performance of the proposed style modeling method using MINE. We conducted two subjective tests, which were the speech quality preference test and the style similarity test. Also for the objective test, we measured speech rate (characters per second) of target speaker and style-adapted speech to show the style similarity. For the experiments, VCTK [47] dataset was applied. VCTK contains 44 hours of clean speech from 108 speakers with text. VCTK has various accents and each speaker

reads about 400 sentences. VCTK is composed of 61 females and 47 males. We downsampled the audio to 22050 Hz for training and trimmed silence in front of speech and after the speech. The window size was 46ms with 11ms hop size. We used 1024 FFT size and 80 channels of mel spectrogram. For d-vector training, we used the same setting as [48] with a 512 d-vector dimension. The configurations of the GST Tacotron model were as follows:

- Text encoder: 512 embedding dimension, pre-net with 256 number of layer and 256 number of units, CBHG module same as [45].
- Decoder: 2 outputs per step, 256 attention dimension, other decoder configuration was same as [45].
- Style token: 10 tokens, multi-head attention with 4 heads, reference encoder was same as in [45].
- Vocoder: Griffin Lim [26] was used as vocoder because of WaveNet training computation.

Also, for MINE network, two feedforward network with [256, 128, 16, 1] number of nodes and we used $\alpha = 0.4$ in Equation 4.5 .We used Adam optimizer in [49] to train our models and we utilized Tensorflow [43], a library for deep learning in our experiments with a single NVIDIA Titan V or NVIDIA Tesla M40 GPU.

For subjective tests, we used 4 test speakers in VCTK, which was randomly chosen with different accents and genders. For both preference tests, we used 40 sentences for each test speaker, which were randomly selected within 100 test sentences. The preference tests were performed with 18 native Korean listeners who are also fluent in English.

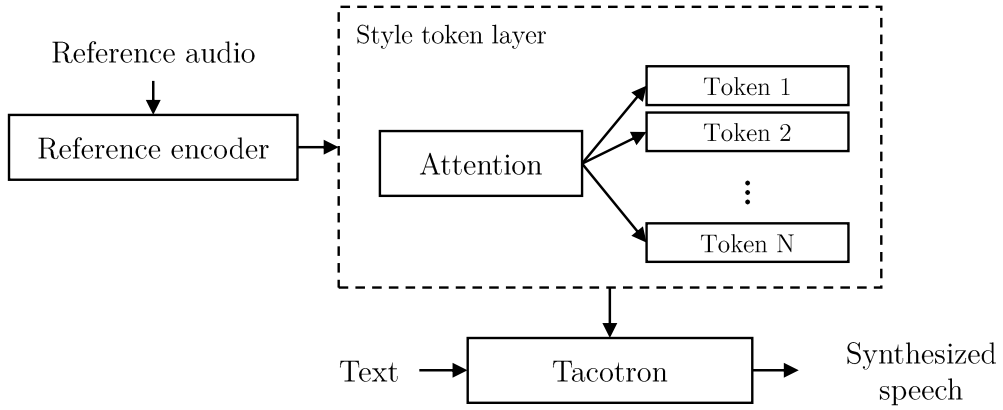
The results of the preference test for style similarity are shown in Figure 4.3b. In every style, our proposed method shows remarkable style similarity than the conventional GST-Tacotron. From these results, we can imply that it is more efficient to use MINE to train target style. Also, the results of preference tests for speech quality are shown in Figure 4.3a. Although there are some differences depending on the style, the overall performance of the proposed method shows better performance. Especially for style 4, there are barely no difference in speech quality evaluation between two methods. Since the accents of style 4 are unusual, it was hard to obtain stable speech and showed relatively low performance than other styles. On the other hand, style similarity in style 4, shows superior performance which means MINE has a strong ability to model target-style even if that style is abnormal.

To measure the speech rate, we used the average speech rate of 40 utterances for each 4 test speakers in VCTK. The results of the speech rate are shown in Figure 4.4. Speech rate of the proposed method is more similar than the conventional GST in every speaker. With the results of the speech rate and the style similarity preference test, we can conclude that MINE-based GST can express the target style more precisely.

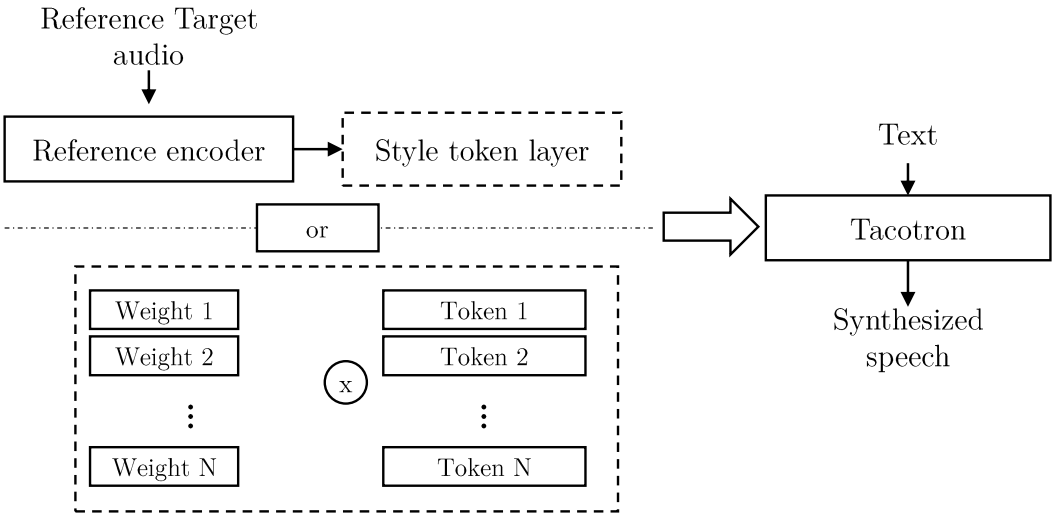
4.5 Summary

In this chapter, we proposed using a MI to style modeling for GST-Tacotron. Utilizing MINE to estimate MI, it was possible to maximize MI in style token layer output with a target-style embedding vector. Within this procedure, disentanglement between style and text information was achieved to make a more controllable style model. The experimental results showed that the proposed style modeling us-

ing MINE approach outperformed in speech quality and style similarity than the conventional GST-Tacotron method.



(a) Training structure of GST Tacotron.



(b) Inference structure of GST Tacotron.

Figure 4.1: The overall structure of GST Tacotron.

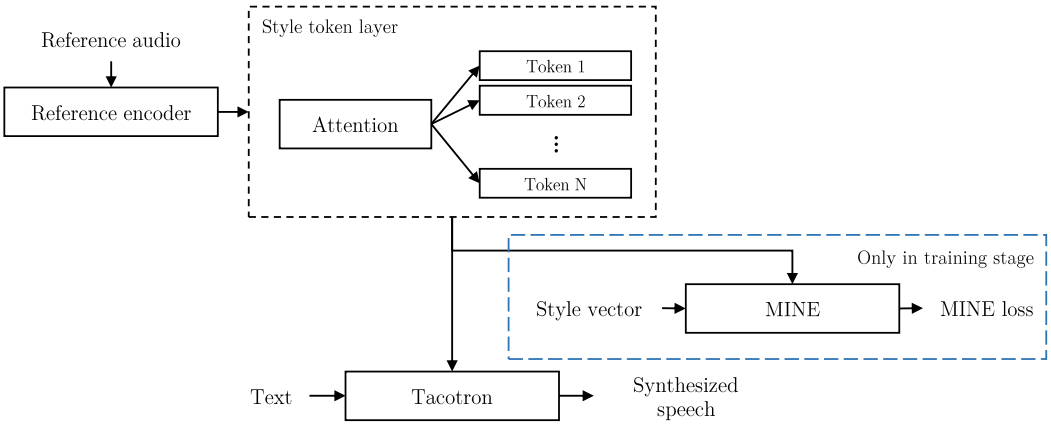
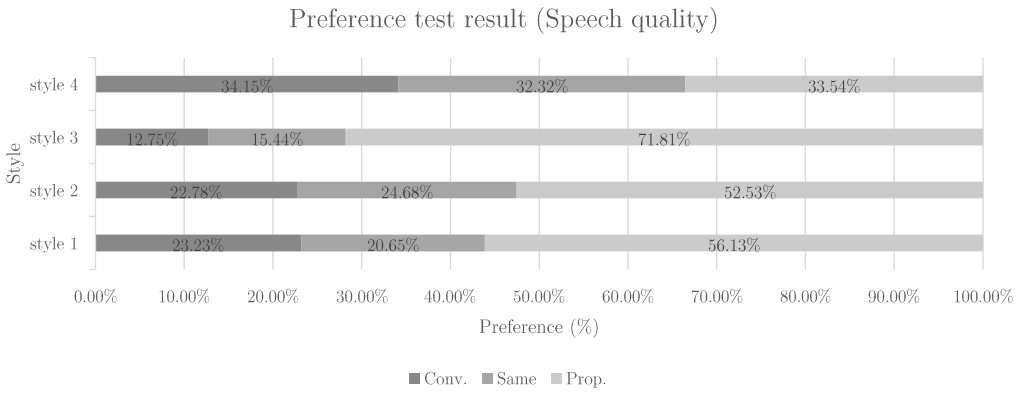
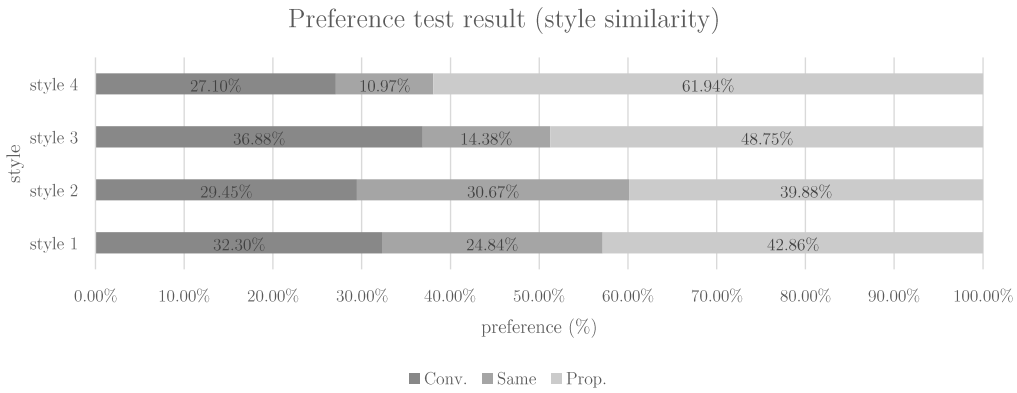


Figure 4.2: The overall structure of the proposed model.



(a) The results of preference test for speech quality.



(b) The results of preference test for style similarity.

Figure 4.3: The results of the subjective tests.

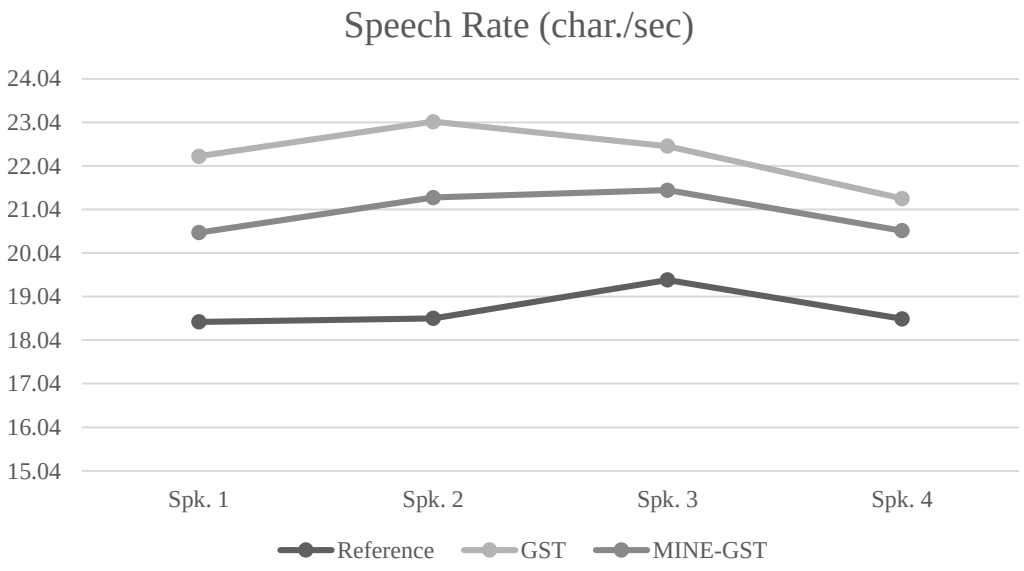


Figure 4.4: The results of the speech rate for style similarity.

Chapter 5

Memory Attention: Robust Alignment using Gating Mechanism for End-to-End Speech Synthesis

5.1 Introduction

A speech synthesis system converts a discrete sequence of text to a sequence of the speech waveform. From the information-theoretic viewpoint, the output sequence contains richer information and is relatively longer than the input sequence. In the majority of end-to-end speech synthesis techniques, an attention mechanism has been proposed to deal with such an imbalance in information. Although the end-to-end speech synthesis system shows better performance than the conventional TTS

system, obtaining a stable and robust attention model is still a challenging task for end-to-end speech synthesis. A mispredicted attention path can result in severe degradation in naturalness and intelligibility of the synthesized speech.

There have been several attempts to make a suitable attention mechanism for speech synthesis. Location sensitive attention (LSA) [8] combines content-based and location-based approaches in attention. Forward attention (FA) [27] achieves attention alignment recursively using a forward algorithm and a transition agent. Dynamic convolution attention (DCA) [29] only utilizes location-relative mechanisms with a dynamic convolution filter. Nevertheless, these attention algorithms still suffer from skipping, repeating, and mumbling of phones, as shown in Figure 5.1. Such problems occur due to the failure in controlling the content (i.e., text embedding sequence and mel-spectrogram sequence) and location information (i.e., previous attention alignment) in the attention mechanism. To achieve a high-quality speech, the alignment path constructed by the attention mechanism should stay or move forward in the time-axis for every decoder time-step monotonically. More specifically, the attention mechanism for speech synthesis should satisfy the following properties:

- *Monotonicity*: To avoid repeating of phones, the alignment path must be monotonically increasing in time for the decoder time-step.
- *Locality*: To prevent skipping of phones, the attention alignment path should be continuous over the input token sequence.
- *Focusing*: To inhibit a mumbling sound, attention should focus on certain input sequence token, and not be distributed over multiple tokens.

The aforementioned attention failures may frequently occur in expressive TTS due to the large variability in target speech.

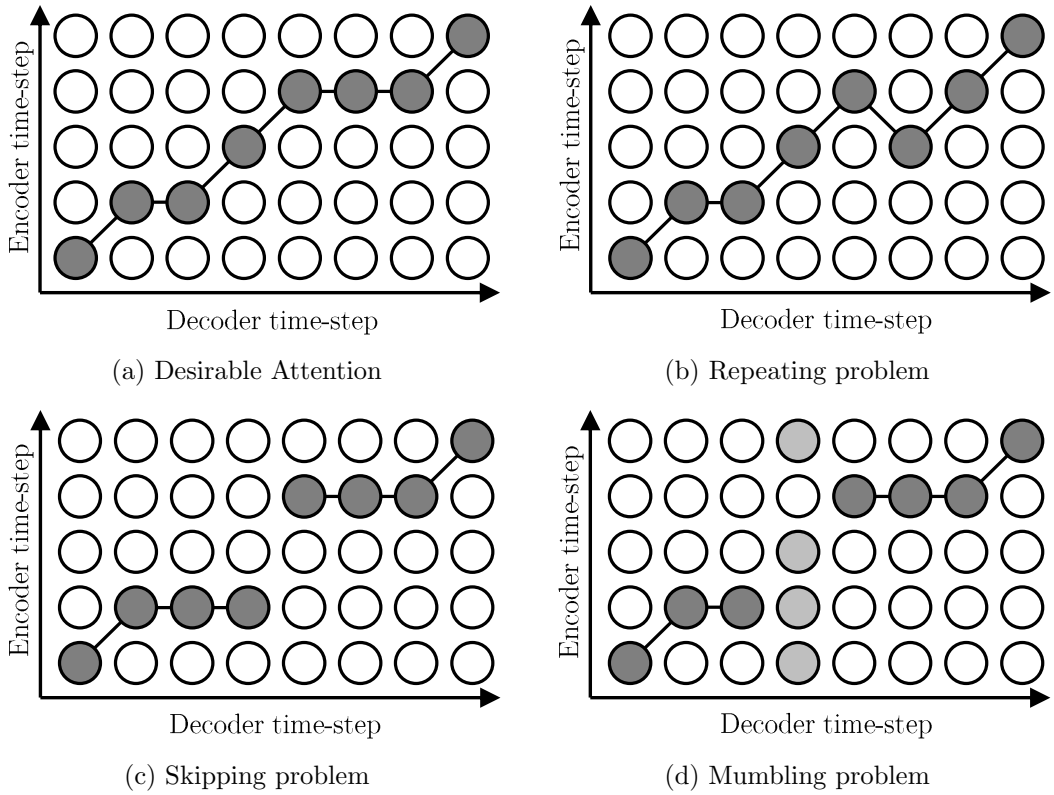
In this chapter, we propose a novel attention algorithm called memory attention which is inspired by a gating technique similar to long-short term memory (LSTM) [18]. LSTM can capture the long-term information of the feature sequence using the memory gates. The memory gates of the LSTM control the amount of information in the input and previous cell states to decide the cell states at the current time-step. We apply a similar gating method to the attention mechanism so that the attention alignment path can also be controlled precisely according to both the location information and content information. With gates associated with the encoder, the memory attention can satisfy the *locality* and *focusing* conditions. The gate associated with the previous alignment is applied to satisfy the *monotonicity* property. Moreover, with the gate related to the decoder, it can help the memory attention to express high-variability in the speech sequence.

The main contributions of this chapter are as follows:

- We propose the memory attention, which is robust against large variability in alignment between the sequence of text and speech signal. Memory attention is motivated by the gating method employed in LSTM, which is a generalized form of the conventional attention mechanism.

- From the experiments, we demonstrate that memory attention can yield better attention alignment than the conventional attention techniques for TTS in both single speaker and emotional speech synthesis cases.

The rest of this chapter is organized as follows. An overview of the conventional attention mechanisms are briefly described in Section 5.2. Then the memory attention is proposed in Section 5.3 and its performance evaluation is provided in Section 5.4. Finally, conclusions are drawn in Section 5.5.



(a) Desirable Attention

(b) Repeating problem

(c) Skipping problem

(d) Mumbling problem

Figure 5.1: Types of the attention failures

5.2 BACKGROUND

In this section, we give a brief review of location sensitive attention (LSA), forward attention (FA), and dynamic convolution attention (DCA).

In the vanilla Tacotron2 system, LSA is used for the attention mechanism [8]. Although LSA is known to yield a more stable alignment path than the fully content-based attention, there still exists a variety of alignment failures at the inference stage. Particularly when training with a highly varying speech database (e.g., emotional speech), we can observe attention failures more frequently. Furthermore, since the convolutional feature of LSA learns a fixed amount of the biased alignment, it is

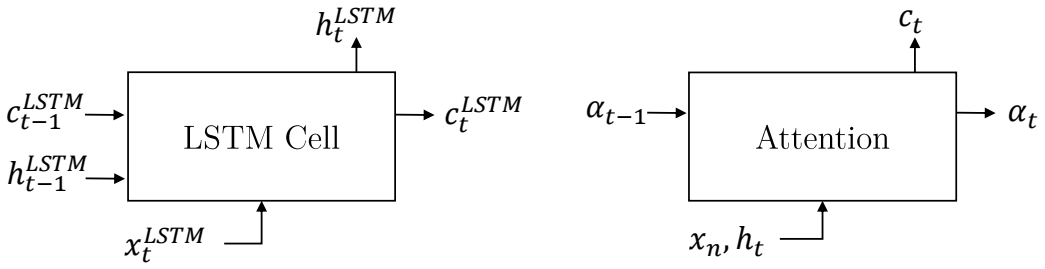


Figure 5.2: The diagram of LSTM and attention mechanism.

more likely to result in a wrong alignment path for rapidly varying speech.

FA [27] is proposed to achieve a better monotonic alignment between the text embedding sequence and the mel-spectrogram sequence than LSA. FA results in better monotonic attention than the LSA, reducing the repeated words of synthesized speech. However, since FA employs the same static convolution filter used in LSA, there still remains the possibility of other types of alignment failures.

DCA [29] is a solely location-relative attention mechanism to generate a long sentence without using content-based terms. DCA shows a better performance, especially in generating long sentence compared to the LSA method while preserving naturalness for shorter in-domain sentence. However, the DCA mechanism has been evaluated only in the single speaker case where the speech characteristics are considered rather homogeneous. Despite the improvement in attention monotonicity, DCA still lacks *focusing* and *locality* since does not contain content-based terms.

5.3 Memory Attention

We propose a novel attention algorithm called memory attention inspired by the memory cell in LSTM. Memory-based recurrent network (e.g., LSTM) and attention

mechanism have a common input-output structure as described in Figure 5.2. In LSTM, the memory cell is controlled by the input, previous memory cell state, and the output with several gates: the input gate, forget gate and output gate. The input gate controls the amount of the input value flowing into the memory, while the output gate handles the quantity of the cell state value remaining to compute the output activation. Also, using the forget gate, the LSTM decides how much to discard its previous cell state. Employing such gating methods, LSTM successfully models a time-varying sequence such as speech.

Similarly, in the attention mechanism, the context vector is controlled by the input text embedding, previous attention alignment, and decoder LSTM state. Also, even though the attention alignment path needs to be modeled sequentially, the conventional attention mechanism has deficient sequential modeling power compared to the gate-based recurrent neural network. Therefore, we adopt a gating technique similar to the LSTM for controlling the attention mechanism to achieve strong sequence modeling power for long time dependency as well as short time dependency. For memory attention, we employ the encoder gate g^{enc}, decoder gate g^{dec}, and update gate g^{up} for the energy function and the alignment. Let $e_{t,n}$ and $\mathbf{h} = [h_1, h_2, ..., h_T]$ denote the energy function on n-th text embedding vector x_n at t-th decoder time-step and decoder LSTM state sequence. Also, $\mathbf{V}_*, \mathbf{W}_*, \mathbf{U}_*$, are weight metrics, $\mathbf{F}^*$ are static convolutional filters, and b_* are biases. The details of our proposed method are as follows:

$$e_{t,n} = v^T \tanh(\mathbf{W}(g^{dec} \odot h_t) + \mathbf{V}(g^{enc} \odot x_n) + b), \tag{5.1}$$

$$g^{dec} = \sigma(\mathbf{U}_d f_t^{dec} + \mathbf{V}_d h_t + b_d), \quad f_t^{dec} = \mathbf{F}^{dec} * \alpha_{t-1}, \tag{5.2}$$

$$g^{enc} = \sigma(\mathbf{U}_e f_t^{enc} + \mathbf{V}_e x_n + b_e), \quad f_t^{enc} = \mathbf{F}^{enc} * \alpha_{t-1}, \tag{5.3}$$

$$g^{up} = \sigma(v_u^T(\mathbf{V}_u h_t + \mathbf{W}_u x_n + \mathbf{U}_u f_{t,n}^{up} + b_u)), \quad f_t^{up} = \mathbf{F}^{up} * \alpha_{t-1}, \tag{5.4}$$

$$\alpha_{t,n}^{'} = \text{softmax}(e_{t,n}), \tag{5.5}$$

$$\alpha_t = g^{up}\alpha_{t-1} + (1 - g^{up})\alpha_t^{'}, \tag{5.6}$$

$$c_t = \sum_{n=1}^{N} \alpha_{t,n} x_n, \tag{5.7}$$

where $\alpha_t^{'}$ is a candidate of current attention alignment at t-th decoder time-step. The memory attention can be interpreted as a generalized form of the content-based attentions as it can express other attention mechanisms with the value of each gate. For $g^{enc} = g^{dec} = g^{up} = 1$, it is equivalent to LSA as in Equation (2.5) to (2.8), while for $g^{enc} = g^{dec} = 1$, it is similar with FA except the normalization process as in Equation (2.9) to (2.12).

With content-related term in Equation (5.1), memory attention can overcome the aforementioned the problem of the solely location-relative attention (i.e., DCA). Memory attention provides location-related features as inputs for the decoder gate, encoder gate, and update gate, rather than applying in the energy function directly.

To encourage the *monotonicity* of the attention, the update gate is formulated as shown in Equation (5.6) to update the alignment path. The update gate plays a similar role in the transition agent in FA as shown in Equation (2.10). It compromises between forgetting the previous alignment and updating the current alignment candidate. If the update gate is set to 0, the alignment path will proceed and completely forget the previous alignment. On the other hand, if the update gate has value 1, the alignment path will stay at the previous alignment. If the monotonicity

is violated by any chance, the update gate can redirect the alignment path to the desirable path by forgetting the previous alignment path.

To deal with *focusing* and *locality* issues, memory attention employs the encoder gate. For each decoder time-step, the encoder gate can discard unimportant input tokens in advance to compute the energy function, which encourages the attention to *focus* on the specific part of the input tokens. In addition, the encoder gate leads the attention to keep the *local* continuity in every encoder time-step by considering the previous attention alignment α_{t-1} to prevent skipping of the input token. Therefore, *focusing* and *locality* problems are mitigated by applying the encoder gate. Furthermore, the decoder gate can regularize the amount of decoder RNN state information applied in the energy function, resulting in robust attention upon a large variety of the speech sequence. Using aforementioned gating methods, we expect the memory attention to learn proper alignments between a text embedding sequence and a mel-spectrogram sequence.

5.4 Experiments

In order to evaluate the performance of the proposed memory attention, we conducted several objective and subjective listening tests for single speaker speech synthesis and emotional speech synthesis. Single speaker speech synthesis was attempted to show the performance of the MA for the general case, and emotion speech synthesis was taken to confirm the robustness of our proposed attention mechanism.

We implemented each model with Tensorflow [43] and it was trained with a single NVIDIA Titan V or NVIDIA Tesla M40 GPU. Also, we utilized the Adam optimizer [41] with the initial learning rate 10^{-3} which was exponentially decayed

after 50,000 steps during training. For the rapid convergence in attention training, we applied the guided attention introduced in DCTTS [12]. For each evaluation, we used 100 test utterances that were not included in the training set as the test data.

For the objective test, we evaluated the word error rate (WER) for single speaker speech synthesis. For emotional speech synthesis, word error count-based error sentence rate (ESR) introduced in Fastspeech [16] was measured to show the amount of alignment failures. As a speech recognition model has poor performance for the emotional speech, we adopted different measures in different scenarios.

For the subjective test, we performed the mean opinion score (MOS) [50] test to evaluate the perceptual quality of the 30 synthesized speech utterances that were randomly selected from the test set. The MOS test was performed with 18 native Korean listeners who are also fluent in English and were asked to score each utterance in the range from 1 to 5: where 1, 2, 3, 4, and 5 each corresponds to bad, poor, fair, good, and excellent. We compared the proposed method, i.e., memory attention (MEM), with the LSA, FA, DCA, and ground truth speech (GT).

5.4.1 Experiments on Single Speaker Speech Synthesis

The objective in these experiments is to compare the performance of the different attention techniques with a typical speech database. For the experiments, we trained all models with a LJ-Speech dataset [51], which contains 13,100 utterances amounting to 23 hours. Each utterance was sampled at 22,050 Hz, and window size was 46ms with 11ms hop size. We used 1,024 FFT size and 80 channels for mel-spectrogram. The structure of Tacotron2 model was almost the same with the vanilla Tacotron2 as in [8], except that instead of using the WaveNet [22], Griffin-Lim vocoder [26] was applied with the CBHG post-filter which converts the mel-spectrogram to the

linear-spectrogram to reduce the training and inference time. The batch size was 16 and trained for 150,000 steps. The details for the Tacotron2 parameters are as follows:

- Text encoder: 512 embedding dimension, three convolution layers with 512 channels, five kernel size, 256 LSTM units for each direction.
- Attention: 128 attention dimension, convolution filter size 32 with 31 kernel size was used for the LSA, FA, and MEM. 64 attention dimension, eight static filters, and eight dynamic filters with length 21, and a length 11 causal prior filter was used for the DCA.
- Decoder: pre-net with 256 number of layers with 256 units, two layers of decoder LSTM with 1,024 units, five convolution layer with 512 channels, and kernel size 5 for postnet.

The Objective Performance Evaluation for Single Speaker

For the objective test, we utilized an end-to-end speech recognition model proposed in [52], which was trained with Librispeech-DB [53]. This speech recognition model obtained 4.92% and 15.15% WER with test-clean and test-other sets. The purpose of using WER is to infer whether there are repeating or skipping words in the synthesized speech. For evaluation, we tested on 100 utterances in the test set. As shown in Table 5.1, memory attention showed lower WERs than the other attention methods. In general, WER is not directly related to the quality of the synthesized speech since the WER does not reflect the naturalness nor sound quality. However, from these results, it can be shown that speech generated by memory attention had better intelligibility and fewer repeating or skipping words. Small batch size (16)

Table 5.1: *Results of the WER [%] for the single speaker case.*

	MEM	MEM ($g^{up} = 0$)	MEM ($g^{dec} = 1$)	LSA	FA	DCA
LJ-Speech	**9.08**	20.79	10.63	11.41	10.57	11.83

was applied compared to the batch size in the original paper (256), accounting for the performance degradation in DCA.

To investigate the influence of each gate in the memory attention, we forced each gate to open or close by manually assigning it to 0 or 1. As text embedding information is crucial to generate speech, it failed to generate natural speech when forcing the value of the encoder gate to 0 or 1. When closing the update gate, since it is difficult to obtain monotonicity, it shows poor performance. As the alignment path cannot proceed when the update gate is opened, it failed to generate natural speech. In the case of the decoder gate, performance degradation is the least when it turns on.

The Subjective Performance Evaluation for Single Speaker

For the subjective test, we chose 30 utterances randomly from the test set. The MOS test results are shown in Table 5.2. The result of the MA outperformed the other methods in the perceptual speech quality. Unlike the WER in Section 5.4.1, DCA scored the second-highest score. This result shows that the DCA had low intelligibility but had better naturalness than the LSA and FA. The MOS test results show that the text embedding sequence and speech sequence are properly aligned through the MA approach and it generated high-quality speech.

Table 5.2: *Results of MOS test with 95% confidence intervals for single speaker case.*

	MEM	LSA	FA	DCA	GT
LJ	**3.735 ± 0.069**	3.554 ± 0.072	3.587 ± 0.073	3.691 ± 0.070	4.883 ± 0.036

Table 5.3: *Results of the word error count-based ESR [%] for emotion speech synthesis case. Re., Sk. represents number of repetition, and number of skipping words*

	MEM			LSA			FA			DCA		
	Re.	Sk.	ESR	Re.	Sk.	ESR	Re.	Sk.	ESR	Re.	Sk.	ESR
ANG	0	1	**1.67**	0	1	**1.67**	1	0	**1.67**	0	3	5
FEA	1	6	**11.67**	3	8	18.33	1	9	16.67	1	13	23.33
JOY	0	0	**0**	1	0	1.67	0	1	1.67	2	3	8.33
NOR	0	1	**1.67**	1	2	5	1	0	**1.67**	1	6	11.67
SAD	1	8	**15**	0	11	18.33	1	11	16.67	1	15	20.33

5.4.2 Experiments on Emotional Speech Synthesis

From the single speaker case, we can verify that our proposed method can express a stable alignment path between input text embedding sequence and output mel-spectrogram sequence. To study the robustness of the attention mechanism upon high variability in speech DB, we evaluated the performance in emotional speech synthesis. Typically, emotional speech has more variation in phone duration or prosody than narrative speech. For emotional speech synthesis, we used an emotional speech database spoken by a single female Korean speaker amounting to 23.8 hours of clean speech. This database is composed of 3,303 utterances of narrative speech (NOR) data and 1,100 utterances each from four different emotions: angry (ANG), fearful (FEA), joyful (JOY), and sad speech (SAD).

Each utterance was sampled at 48,000 Hz but downsampled to 22,050 Hz, and window size was 46ms with 11ms hop size. We used 1,024 FFT size and 80 channels of mel-spectrogram, the same as the single speaker. The batch size was eight and trained for 200,000 steps. As the average audio time length of the emotional database

Table 5.4: *Results of the MOS test with 95% confidence intervals for emotional speaker case.*

	MEM	LSA	FA	DCA	GT
ANG	**4.020 ± 0.069**	3.913 ± 0.069	3.722 ± 0.074	3.954 ± 0.068	4.764 ± 0.055
FEA	**3.680 ± 0.075**	3.406 ± 0.085	3.476 ± 0.086	3.083 ± 0.088	4.567 ± 0.069
JOY	**4.233 ± 0.061**	4.091 ± 0.064	4.143 ± 0.063	3.822 ± 0.075	4.781 ± 0.049
NOR	**4.198 ± 0.058**	3.985 ± 0.070	4.056 ± 0.067	3.646 ± 0.077	4.511 ± 0.074
SAD	**3.856 ± 0.070**	3.420 ± 0.083	3.420 ± 0.095	3.050 ± 0.092	4.464 ± 0.074

(11sec) is longer than the single speaker database (6.3sec), we utilized a smaller model size of baseline Tacotron2 as well as smaller batch size. The structure of emotional speech synthesis is shown in Figure 5.3. We used one-hot encoding to input emotion, with a look-up table-based emotion embedding layer. The output of the emotion embedding layer is conditioned on Tacotron2. For conditioning, we applied gated linear units (GLU) [54] as follow:

$$h_{cond} = (\mathbf{W}_1 h_1^{split} + \text{softmax}(\mathbf{V}_1 E + b_e) + b_1) \otimes \sigma(\mathbf{W}_2 h_2^{split} + V_2 E + b_2), \quad (5.8)$$

where h_*^{split} and E are each half-split output of the encoder CNN and emotion embedding output. In this experiment, we added additional emotion embedding term V_2E for gate control. As in Section 5.4.1, Griffin-Lim vocoder and CBHG post-filter were applied instead of WaveNet.

The other detailed configurations of Tacotron2 model are as follows:

- Text encoder: 256 embedding dimension, three convolution layers with 512 channels, five kernel size, 128 LSTM units for each direction.

- Attention: 64 attention dimension, convolution filter size 32 with 31 kernel size was used for the LSA, FA, and MEM. 64 attention dimension, eight static

filters, and eight dynamic filters with length 21, and a length 11 causal prior filter was used for the DCA.

- Decoder: pre-net with 128 number of layers and 256 units, two layers of decoder LSTM with 512 units, five convolution layers with 512 channels, and kernel size 5 for postnet.

The Objective Performance Evaluation for Emotional Speech

For the objective test, we counted the number of word repeating and word skipping errors as in Fastspeech [16]. Then, we counted sentence errors to compute ESR. To evaluate the performance, we used 50 sentences that were randomly selected from the test set for each emotion. The ESR results are shown in Table 5.3, which demonstrates that the memory attention can alleviate the attention failures more appropriately than the other attention techniques. Typically, ANG has very short speech duration, which does not require high expressiveness in attention path. Therefore, ANG showed similar performance results among all the attention mechanisms. On the other hand, as FEA and SAD have many pauses, breathing sound, and quivering voice compared to the other emotions, their synthesized speech quality was degraded.

The example of attention failure is shown in Figure 5.4. As shown in Figure 5.4a, the alignment paths of LSA and DCA were disconnected which led to mispronunciation. The alignment of the FA seems normal, but attention stayed focus on one particular input token for long decoder time-step which yielded bad naturalness. Similarly, in Figure 5.4b, LSA failed to focus on specific input tokens that result in mumbling sounds. Also, FA had a missing path and DCA had both problems. From

the ESR results and its example, it can be shown that our proposed method showed outstanding performance in attention robustness.

The Subjective Performance Evaluation for Emotional Speech

For the MOS test in the emotional speech case, we used 30 utterances randomly chosen from the test set for each emotion. The MOS results shown in Table 5.4 reveal that the proposed method outperforms the other methods in terms of the perceptual quality of the synthesized emotional speech. As in the objective test, FEA and SAD cases show lower scores than the other emotions, but the relative improvement in performance between the memory attention and the other attention methods increased. These results show that the memory attention provided robust attention, leading to better synthesized speech quality, especially in FEA and SAD.

The DCA showed relatively good performance than LSA and FA for the single speaker but showed the worst performance in emotional speech synthesis. Since the DCA does not utilize the embedding sequence in the energy function, it seems to fail to adapt emotion differences in the attention model. The FA performed slightly better than LSA except for the ANG case, as in the single speaker.

5.5 Summary

In this chapter, we proposed memory attention as a novel technique for robust attention when the data contains various durations such as emotional speech. The proposed approach was inspired by the gating technique in LSTM. In the memory attention, the attention alignment path and context vector were controlled sophisticatedly with gates associated with the previous alignment, input text embedding

sequence, and generated mel-spectrogram sequence. The memory attention was a generalized form of the content-based attention and can redirect the attention path flexible, which leaded to making a robust attention path in unexpected situations. From the experimental results, it can be shown that our proposed method outperformed conventional LSA, FA, and DCA.

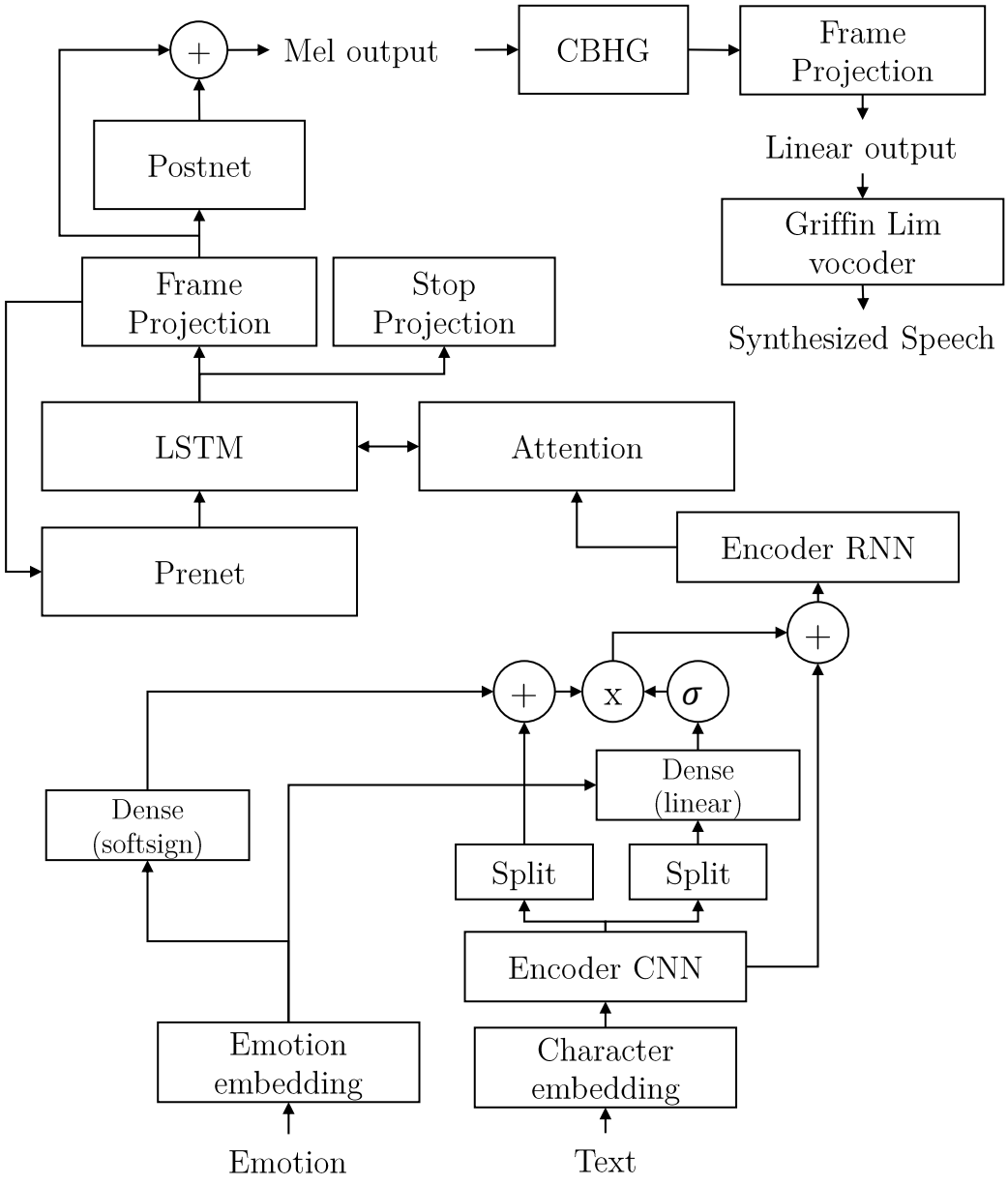
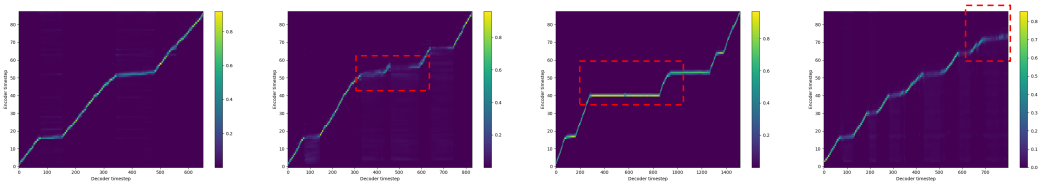
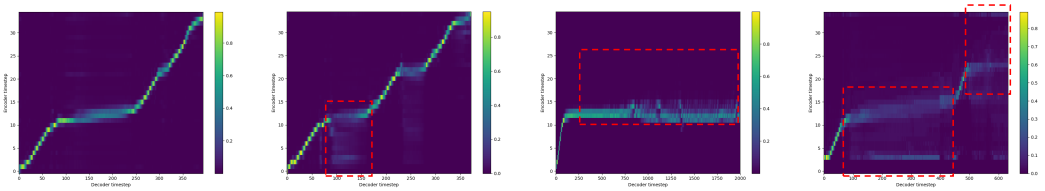


Figure 5.3: The structure of the emotional speech synthesis experiments.



(a) The alignment path example of fear emotion in MEM, LSA, FA and DCA (from left to right).



(b) The alignment path example of sad emotion in MEM, LSA, FA and DCA (from left to right).

Figure 5.4: Comparison of alignment path of fear and sad emotion. For each alignment path, we used same sentence and different emotion.

Chapter 6

Selective Multi-attention for style-adaptive end-to-End Speech Synthesis

6.1 Introduction

To achieve human-like speech quality, the speech synthesis system should express various speaking styles such as emotion and prosody. Also, multi-speaker TTS is required for a personalized TTS which is an essential aspect for many applications. There have been several attempts to adopt multi-speaker characteristics and style to end-to-end speech synthesis. In [48], speaker embedding vectors with a fixed length such as d-vector is used to transfer speaker characteristics. Moreover, style modeling methods that adapt the speaking style of a reference audio clip to the end-to-end speech synthesis system have been developed. The reference encoder-based approaches [44], [55] summarize the speaking style of a reference audio and reflects

it to the synthesized speech. The global style token [45] extracts speech style and replicates the speaking style of a reference audio clip using a weighted sum of the style tokens. These speaker embedding and style transfer techniques are implemented mostly on an attention-based end-to-end speech synthesis system. However, despite its success in some style-adaptive speech synthesis, due to the unstable alignments among different styles, the quality of the synthesized speech is still low compared to the single speaker model with a monotonous speech.

The attention-based alignment in the style-adaptive speech synthesis has two significant challenges. One is to have a monotonic alignment path without an attention failure, and the other is to have a diverse attention alignment depending on the speaking styles. Current research in the attention mechanism focuses on mitigating the attention failure which results in the repetition, skipping, and murmuring words of the synthesized speech [27], [56] with single attention model. These attempts show performance improvement in generating a monotonic alignment path, but typically prove its performance with the monotonous speech data rather than the speech data with various styles. As a result, these approaches may have limitations in representing various alignment paths according to the speech styles. For this reason, in the style-adaptive end-to-end speech synthesis, some attention paths fail to align a proper path and result in generating uncomfortable and unstable speech. Especially in the outlier style cases, using single attention shows poor performance.

To avoid such lack of expressiveness in alignment, we propose selective multi-attention, which uses multiple attentions for the style-adaptive end-to-end speech synthesis instead of a single attention. Multi-attention is applied to generate diverse attention alignment candidates. Also, we adopt a selection network that learns to select an appropriate attention model for the target style within multiple attentions.

The selection network utilizes a softmax output layer to make a soft decision given style embedding vector. The main contributions of this chapter are as follow:

- We propose the selective multi-attention, which can express diverse alignment path depending on styles.
- The selective multi-attention can be applied in various single attention models.
- From the experiments, we demonstrate that the selective multi-attention can yield better attention alignment than the conventional attention techniques for TTS in both multi-speaker and emotional speech synthesis cases.

In Section 6.2 we briefly review the conventional attention mechanisms. The detailed structure and training scheme of our proposed method is introduced in Section 6.3. Experimental results based on Tacoton2 systems and discussion are presented in Section 6.4, and finally, we make a conclusion in Section 6.5.

6.2 BACKGROUND

Conventional attention model such as Location-sensitive attention (LSA) [20], Forward attention (FA) [27], dynamic convolutional attention (DCA) [29] is proposed for single narrative speaker. Moreover, FA and DCA have been evaluated only in the single speaker case where the speech characteristics are considered rather homogeneous. The main purpose of the aforementioned attention mechanisms is to achieve a stable monotonic attention path without skipping or repeating words in an utterance. However, these attention techniques do not pay much attention to making diverse alignments according to the various styles.

There is another steam of attention mechanism in speech synthesis such as multi-head attention (MHA) [57]. MHA is used widely in self-attention model such as Transformer-based TTS [13] which is non-autoregressive model. MHA split query, key, value of attention and compute attention separately, and merge afterward. Since MHA split attention, each splitted small attention can have little expressiveness on dynamic attention path. From our preliminary experiments, we have found that, MHA is difficult to train style-adaptive speech synthesis when applied in the Tacotron-based model.

6.3 Selective multi-attention model

As mentioned in the previous section, single attention models in the style-adaptive speech synthesis are focused on making a stable alignment path instead of generating a dynamic attention path according to the style. To alleviate this problem, we use K multiple attentions to enhance the expressiveness of the attention model. By using multiple attentions, we can obtain multiple candidates of the alignment paths. To pick the most suitable alignment path for the target style among these candidates, we propose to use a style-based selection network which make a soft decision among multiple attentions. Utilizing soft decision, a selection network can choose an alignment path that can express the target style. The selection network is composed of simple feedforward layers with a softmax activation as an output layer. The number of output layer nodes in the selection network is the same as the number of attention model K in the multi-attention model. Using the selection network, the selective multi-attention model can be derived as follow:

$$\alpha_t = \sum_{k=1}^{K} \mathbf{S}_k \alpha_{t,k}, \quad \alpha_{t,n,k} = softmax(e_{t,n,k}), \tag{6.1}$$

$$c_t = \sum_{k=1}^{K} \mathbf{S}_k c_{t,k}, \quad c_{t,k} = \sum_{n=1}^{N} \alpha_{t,n,k} x_n, \tag{6.2}$$

where index k denotes k-th attention model in multi-attention model and $\mathbf{S}$ indicates the output layer of a selection network. As shown in Equation 6.1, we can use the attention context vector from any attention mechanisms for selective multi-attention. From this point, selective multi-attention can be expanded to other attention mechanisms. In this chapter, we applied the LSA as in vanilla Tacotron2 to show the effectiveness of our method.

6.4 EXPERIMENTS

To evaluate the performance of the proposed selective multi-attention model (SMA) in the speech synthesis system, a subjective listening tests were conducted for the multi-speaker speech synthesis and the emotional speech synthesis. For both experiments, we used the mean opinion (MOS) [50] test to evaluate the perceptual quality of synthesized speech, and style similarity preference test to compare the ability to express diverse alignment paths from different attentions. The MOS test was performed with 17 native Korean listeners who were also fluent in English and were asked to score in the range from 1 to 5: where 1, 2, 3, 4, and 5 each correspond to bad, poor, fair, good, and excellent.

The similarity preference test was also performed with the same listeners and was asked to choose the similarity of two different methods with reference style. Every

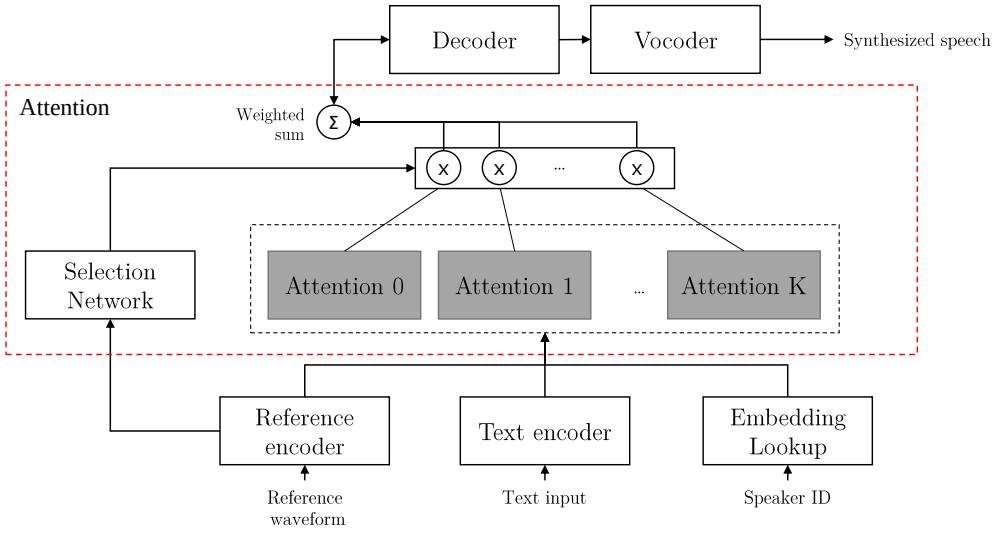


Figure 6.1: Overall structure of selective multi-attention multi-speaker model

subjective listening test was performed with randomly chosen sentences among 100 test sentences which were not included in the training process. We applied Adam optimizer in [49] to train our model and we used Tensorflow [43], a library for deep learning, in our experiments with a single NVIDIA Titan V or NVIDIA Tesla M40 GPU.

6.4.1 Multi-speaker speech synthesis experiments

For the multi-speaker speech synthesis, we applied the reference encoder-based multi-speaker speech synthesis system as in [44]. The reference encoder summarizes reference speech's prosody and conditioned on Tacotron2 model. The architecture of the reference encoder was consists of 6-layer convolutional network followed by a single layer zoneout LSTM. The convolutional network had composed of 3×3 filters with 2×2 strides, and the number of filters are 32, 32, 64, 128, and 128. The zoneout LSTM was uni-directional with 128-dimensional output. The overall procedure of

	p236	p237	p247	p262	p271
Gender	Female	Male	Male	Female	Male
Accents	English	Scottish	Scottish	Scottish	Scottish
	p298	p318	p330	p343	p345
Gender	Male	Female	Female	Female	Male
Accents	Irish	American	American	Canadian	American

Table 6.1: *Gender and accent of test speakers.*

the SMA multi-speaker model is in the Figure 6.1. For other attentions, we simply replaced SMA with single attention model in Figure 6.1.

For the multi-speaker experiments, we trained each model with public multi-speaker database VCTK [47] which contains 44 hours of clean speech from 108 speakers with scripts. VCTK has various accents and each speaker reads about 400 sentences. VCTK is composed of 61 females and 47 males. We downsampled the audio to 22050 Hz for training, and trimmed silence in front of speech and after the speech. The window size was 46ms with 11ms hop size. We used 1024 FFT size and 80 channels of mel spectrogram. For the subjective test, we tested on 10 test speakers in VCTK with information in Table 6.1 which was randomly chosen with different accents and gender. For the MOS test and speaker similarity preference test, we utilized 15 sentences for each test speaker within 100 test sentences.

We compared the proposed SMA to LSA, FA, which are the single attention as aforementioned in Section 6.2, and ground truth speech (GT). For SMA, we used 2, 4, and 16 attentions, denoted as SMA2, SMA4, and SMA16. Also, the selection network in SMA was consisted of 4 feedforward hidden layers with 512 ReLU activation nodes and an output layer with softmax activation nodes.

The configurations of the Tacotron 2 model were as follows:

- Text encoder: 512 embedding dimension, three convolution layers with 512

Table 6.2: *Results of MOS test with 95% confidence intervals for single speaker case.*

method	SMA2	SMA4	SMA16
MOS	3.4086 ± 0.0825	3.4416 ± 0.0980	**3.6011 ± 0.0786**
method	LSA	FA	GT
MOS	3.1827 ± 0.1114	3.3663 ± 0.0956	4.8553 ± 0.0429

channels, 5 kernel size, 256 bidirectional LSTM units for each direction.

- Attention: 128 attention dimension, convolution filter size 32 with 31 kernel size(in case of SMA16, we used 64 attention dimension because of the limitation of GPU memory).

- Decoder: pre-net with 256 number of layer and 256 number of units, two layer of decoder LSTM with 1,024 units, five convolution layer with 512 channels and kernel size 5 for postnet.

- Vocoder: different from original Tacotron 2 we use Griffin Lim [26] as vocoder because of WaveNet training computation.

- Post-filter: different from original Tacotron 2 we apply CBHG based post-filer to convert mel spectrogram to linear spectrogram to improve the computational time for training and inference.

MOS test evaluation

The results of the MOS test are represented in Table 6.2. We averaged the MOS test results of 10 test speakers. The MOS test results show that SMA16 has the best performance among other attentions. As we used Griffin-Lim vocoder instead of the neural vocoder (e.g., WaveNet), the overall performance of the experiments is degraded than that of utilizing the neural vocoder. Within three different SMAs,

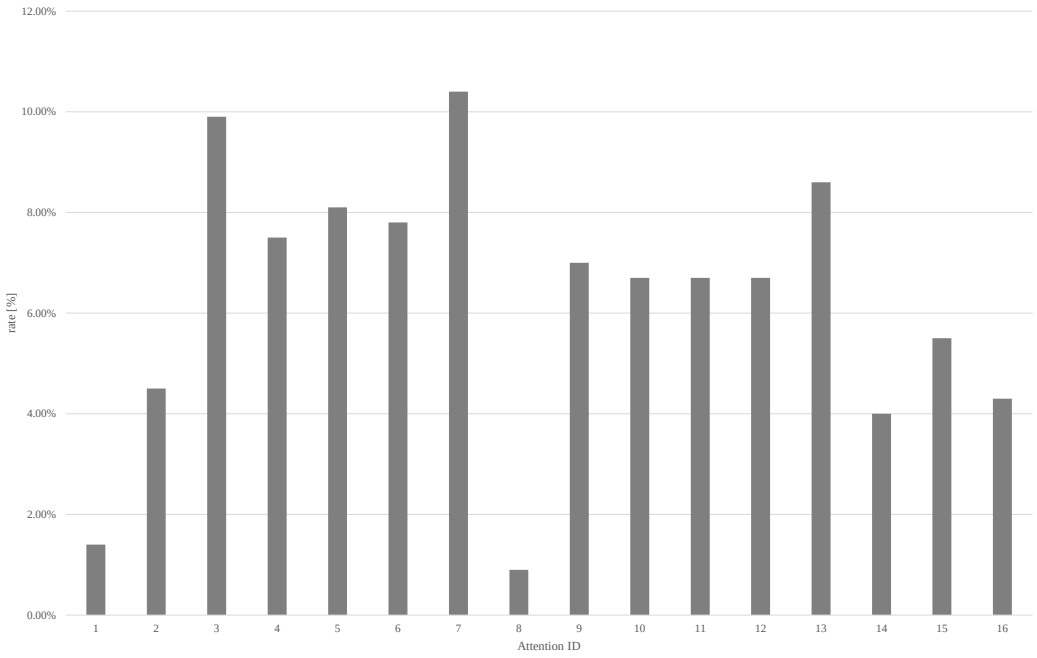
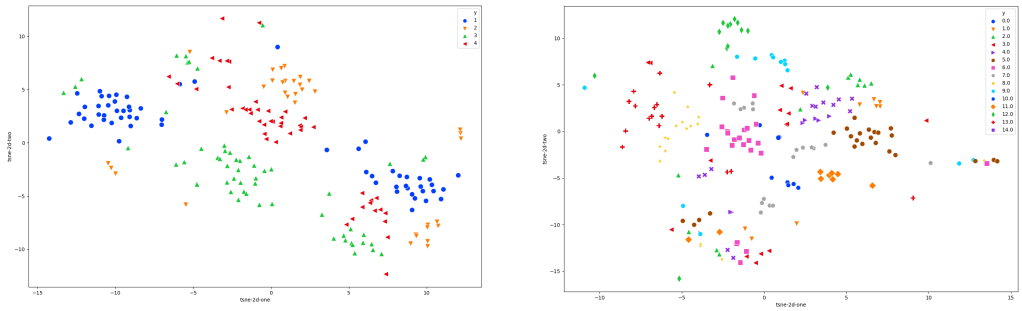


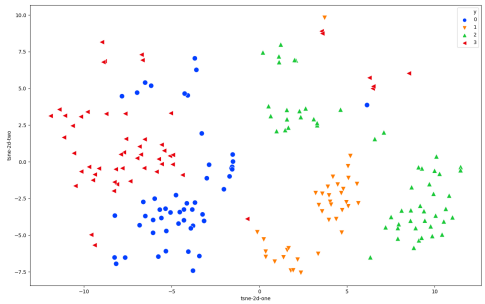
Figure 6.2: Attention selection rate of multi-attention in SMA16.

SMA16 has shown the best quality. To demonstrate the alignment variability of SMA, we calculate the attention selection rate as in Figure 6.2, which estimates how much each attention in multi-attention is selected. This result shows that the selection network of SMA selects diverse alignment paths within candidates. Also, to investigate the tendency of selected attention depending on 200 different reference speeches, we visualize the attention selection according to the reference embedding using t-distributed stochastic neighbor embedding (t-SNE) [58] with fixed speaker. The t-SNE is a popular data visualization method that projects high dimensional data into a smaller dimension (e.g., 2-dimensional one). We projected the reference embedding vectors, which was extracted with different reference speech, into the two-dimensional space in Figure 6.3 for the case of SMA4 and SMA16. Note that the color and shape of the dot represent selected attention according to the reference

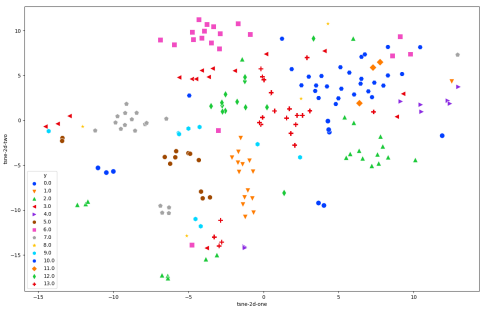


(a) SMA4 (speaker 1)

(b) SMA16 (speaker 1)



(c) SMA4 (speaker 2)



(d) SMA16 (speaker 2)

Figure 6.3: Visualization of attention selection using t-SNE in the multi-speaker d-vector.

embedding vector. In both cases, it seems the selected attention is well separated according to style. As some attentions were barely selected in SMA16, the best number of attention needed for VCTK seems to be between 4 and 16.

Similarity test evaluation

The results of the similarity preference tests are presented in Figure 6.4. From each comparison, SMA16 shows relatively similar style representation, among other attentions. From these results, we can verify that SMA can express stylish speech more properly than other attention methods.

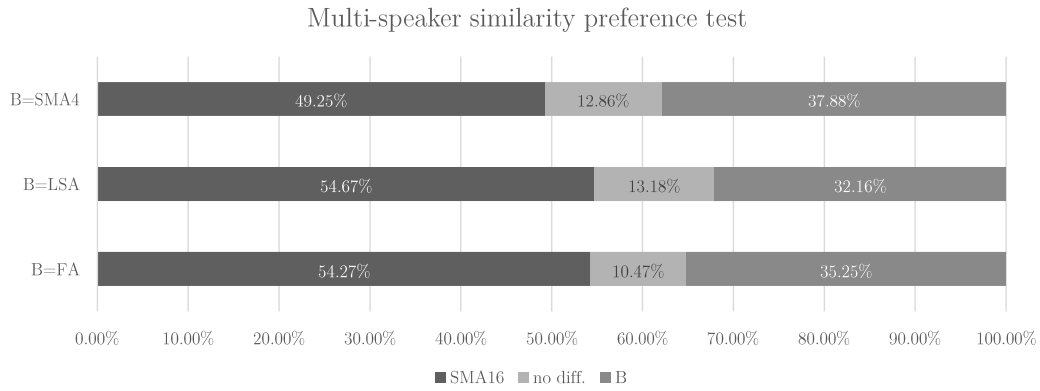


Figure 6.4: The results of similarity preference test in multi-speaker speech synthesis.

6.4.2 Experiments on Emotional Speech Synthesis

The objective of the emotional speech synthesis experiments is to confirm the performance of SMA upon large variability in speech DB. Typically, emotional speech has more variation in phone duration or prosody than narrative speech. For the construction of emotional speech synthesis, we used an emotional speech synthesis database spoken by a single female Korean professional voice actress which contains 23.8 hours of clean speech. This database is composed of 3,303 utterances of narrative speech (NOR) data and 1,100 utterances each from four different emotions: angry (ANG), fearful (FEA), joyful (JOY), and sad speech (SAD).

Each utterance was sampled at 48,000 Hz but downsampled to 22,050 Hz, and window size was 46ms with 11ms hop size. We used 1,024 FFT size and 80 channels of mel-spectrogram, the same as the multi-speaker. The batch size was eight and trained for 200,000 steps. The structure of emotional speech synthesis is shown in Figure 6.5. We applied a one-hot encoding scheme to input emotion, with a look-up table-based emotion embedding layer. The output of the emotion embedding layer is conditioned on Tacotron2. For conditioning, we employed gated linear units

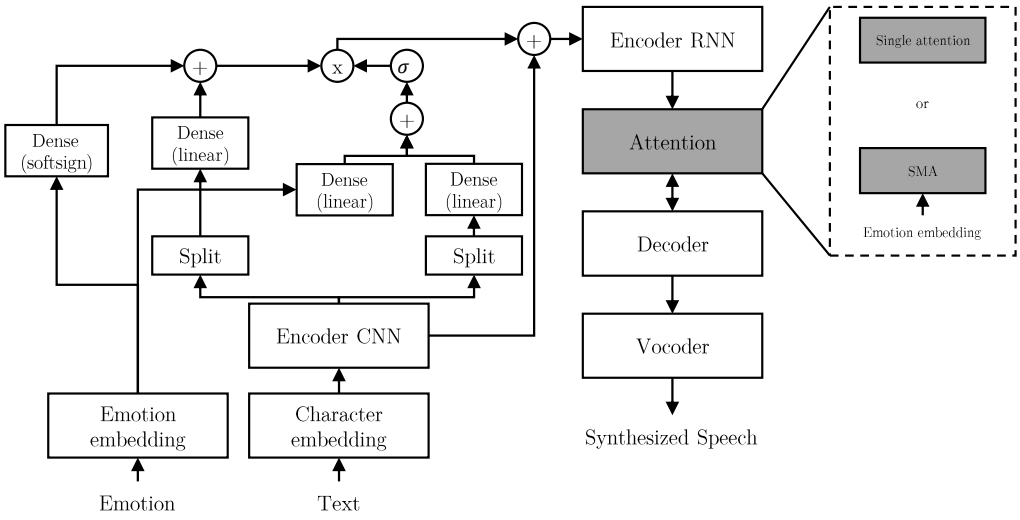


Figure 6.5: The overall structure of emotional speech synthesis.

(GLU) [54] as follow:

$$h_{cond} = (\mathbf{W}_1 h_1^{split} + \text{softsign}(\mathbf{V}_1 E + b_e) + b_1) \\ \otimes \sigma(\mathbf{W}_2 h_2^{split} + V_2 E + b_2), \tag{6.3}$$

where h_*^{split} and E are each half-split output of the encoder CNN and emotion embedding output. In this experiment, we applied additional emotion embedding term $V_2 E$ for gate control. As in Section 6.4.1, Griffin-Lim vocoder and CBHG post-filter were applied instead of WaveNet. The configurations of the Tacotron 2 model were as follows:

- Text encoder: 256 embedding dimension, three convolution layers (512 channels, 5 kernel size), 128 bidirectional LSTM units for each direction.

- Attention: 64 attention dimension, convolution filter size 32 with 31 kernel size.

- Decoder: pre-net with 128 number of layer and 256 number of units, two layer of decoder LSTM with 512 units, five convolution layer with 512 channels and kernel size 5 for postnet.

- Vocoder: different from original Tacotron 2 we use Griffin Lim as vocoder because of WaveNet training computation.

- Post-filter: different from original Tacotron 2 we apply CBHG based post-filer to convert mel spectrogram to linear spectrogram to improve the computational time for training and inference.

For subjective tests, we tested on 20 sentences for each emotions which were randomly chosen within 100 test sentences. We compared the proposed SMA to LSA, FA, MHA with 2 head and ground truth speech (GT). For SMA, we utilized 2, 4 attentions fewer than multi-speaker case since there are only five emotions. Also, the selection network in SMA was consisted of 4 feedforward hidden layers with 512 ReLU activation nodes and an output layer with softmax activation nodes.

MOS test evaluation

The results of the MOS tests are shown in table 6.3. From these results, SMA2 shows the best performance compared to other attention methods. As FEA and SAD have more sobering sound, breathing and unstable speech, it has lower quality than other emotions. However, the performance improvement of these two emotions using SMA is more significant than the others. From these results, it can be seen that SMA can synthesize better across the diverse style. Since there are only five emotions, using just two attentions is enough to express diverse alignment path than utilizing four attentions. Moreover, using four attention for five emotions provoke

Table 6.3: *The results of the MOS test with 95% confidence intervals for emotional speaker case.*

	SMA2	SMA4	FA	LSA	MHA	GT
ANG	**4.124** ± **0.072**	3.947 ±0.078	4.026 ±0.076	3.953 ±0.077	3.915 ±0.083	4.859 ±0.040
FEA	**3.241** ± **0.085**	3.056 ±0.095	3.103 ±0.096	3.062 ±0.093	3.015 ±0.097	4.706 ±0.062
JOY	**4.115** ± **0.076**	3.979 ±0.075	3.965 ±0.080	4.024 ±0.071	3.971 ±0.080	4.797 ±0.047
NOR	**4.003** ± **0.071**	3.859 ±0.087	3.932 ±0.079	3.874 ±0.082	3.785 ±0.083	4.803 ±0.055
SAD	**3.350** ± **0.079**	3.112 ±0.092	3.147 ±0.091	3.079 ±0.092	3.221 ±0.082	4.776 ±0.058

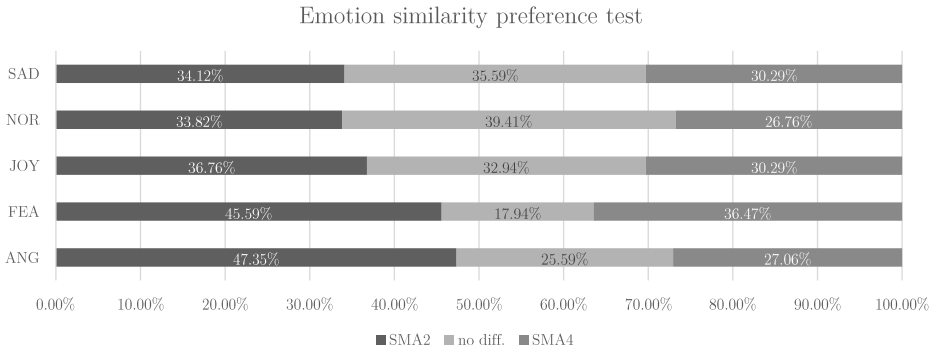
insufficient training which caused the poor performance. In our experience, MHA using the number of head bigger than two fails to have a stable attention alignment path, which means it is inadequate to use in style-adaptive speech synthesis.

Similarity test evaluation

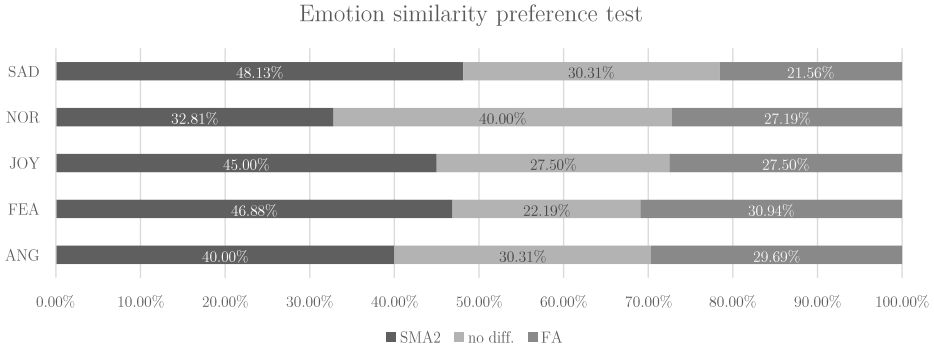
For the similarity test evaluation, we compared SMA2 with FA, MHA, and SMA4, respectively. From the results in Figure 6.6, SMA2 outperformed than any other attentions. From these results, we can observe that using SMA can make a proper alignment path for emotional speech synthesis. Since SMA2 used two attentions in the emotional case, the response to 'no difference' is many compared to the multi-speaker model which utilized 16 attentions. However, for every case compared to SMA2, the results show better emotion similarity and prove that it is more effective for emotional speech synthesis.

6.5 Summary

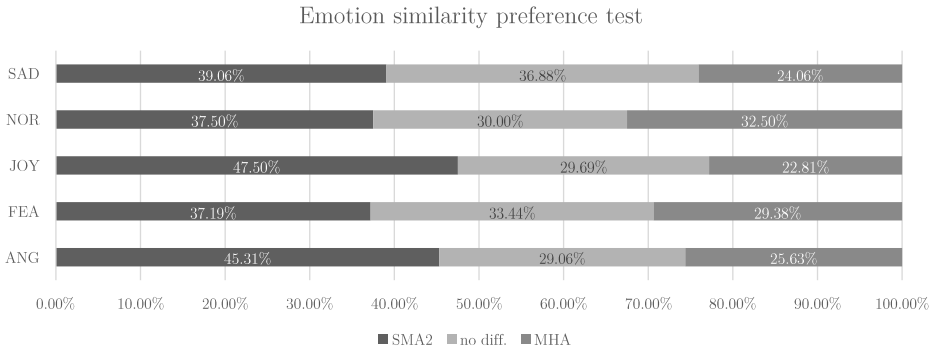
In this chapter, we proposed using a selective multi-attention model for style-adaptive end-to-end speech synthesis system. Since style characteristic in speech contains high variability, instead of applying a single attention model, we utilized a multi-attention model to make candidates of alignment path depending on the style. Also, to select the multi-attention models, we used a selection network to make a soft decision of which attention is appropriate to the target style. As a results, selective multi-attention can have proper alignment path for diverse speaking styles. Also, selective multi-attention can be applied to any single attention mechanism. From the experimental results, it was shown that the proposed selective multi-attention based method outperformed in speech quality and style similarity than the conventional single attention method for multi-speaker and emotional end-to-end speech synthesis.



(a) SMA2 vs. FA



(b) SMA2 vs. MHA



(c) SMA2 vs. SMA4

Figure 6.6: The results of similarity preference test for emotional speech synthesis.

Chapter 7

Conclusions

This dissertation proposed some of the solutions which resolve the drawbacks of the conventional neural network-based speech synthesis system. In neural SPSS, the modeling power of the conventional long short term memory (LSTM)-based approach seemed to be inefficient due to the deterministic property. In style-adaptive end-to-end speech synthesis, conventional unsupervised style modeling such as global style token (GST) was hard to focus on the target style specifically. Also, attention mechanisms in the conventional end-to-end methods do not consider dynamic style variation of speech. To overcome these problems, novel methods for neural network-based speech synthesis were proposed.

Firstly, we proposed using a variational RNN (VRNN)-based method as an alternative method for the acoustic modeling in neural statistical parametric speech synthesis system. By applying the VRNN, the acoustic model can express the variability efficiently within the highly structured data. Also, instead of using the conventional variational lower bound, we utilized an adversarial training scheme to increase the dynamic range for synthesized speech data. We called this VRNN with an adversar-

ial training scheme as AdVRNN. From the experimental results, it was shown that the proposed AdVRNN based method outperformed the conventional RNNs-based method for acoustic modeling.

Secondly, we proposed using mutual information (MI) to style modeling for GST-Tacotron. Due to the conventional GST-Tacotron was modeled using unsupervised training, it was difficult to train a specific target style. Utilizing mutual information neural estimator (MINE) to estimate MI, it was possible to maximize MI of style token layer output and target style embedding vector. Also, within this procedure, disentanglement between style and text information was achieved to make a more controllable style-adaptive model. The experimental results showed that the proposed style modeling using the MINE approach outperformed the conventional GST-Tacotron method in speech quality and style similarity.

Thirdly, we proposed memory attention, a novel technique for robust attention alignments when the data contains various durations. The proposed approach was inspired by the gating technique in LSTM. In the memory attention, the attention alignment path and context vector were controlled sophisticatedly with gates associated with the previous alignment, input text embedding sequence, and decoder-state sequence. The memory attention is a generalized form of the conventional content-based attentions and can redirect the attention path flexible, which led to making a robust attention path in unexpected situations. The experimental results showed that the proposed method outperformed the conventional attention mechanisms in style-adaptive speech synthesis.

In the last approach, we proposed to use a selective multi-attention model (SMA) for style end-to-end speech synthesis systems. Since style characteristic in speech contains high variable speech duration, instead of applying a single attention model,

we utilized a multi-attention model to make candidates of the alignment path depending on the style. Also, to select the proper attention within multiple attentions, we used a selection network to decide which attention was appropriate to the target style. As a result, SMA can have a proper alignment path for diverse speaking styles. The experimental results showed that the proposed SMA method outperformed the conventional single attention method for multi-speaker and emotional end-to-end speech synthesis in speech quality and style similarity.

Bibliography

[1] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, May 1996, pp. 373–376.

[2] A. W. Black and P. Taylor, "CHATR: a generic speech synthesis system," in *Proc. COLING*, 1994, pp. 983–986.

[3] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, vol. 3, 2000, pp. 1315–1318.

[4] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. ISCA SSW6*, 2007, pp. 294–299.

[5] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2013, pp. 7962–7966.

[6] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[7] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. Saurous, "Tacotron: Towards end-to-end speech synthesis," 08 2017, pp. 4006–4010.

[8] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017. [Online]. Available: http://arxiv.org/abs/1712.05884

[9] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," *CoRR*, vol. abs/1702.07825, 2017. [Online]. Available: http://arxiv.org/abs/1702.07825

[10] S. Ö. Arik, G. F. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *CoRR*, vol. abs/1705.08947, 2017. [Online]. Available: http://arxiv.org/abs/1705.08947

[11] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *CoRR*, vol. abs/1710.07654, 2017. [Online]. Available: http://arxiv.org/abs/1710.07654

[12] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4784–4788.

[13] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[14] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=SkFAWax0-

[15] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. C. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *ICLR*, 2017.

[16] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Proceedings of the Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[18] A. Graves, "Supervised sequence labelling," in *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13.

[19] J. Libovickỳ and J. Helcl, "End-to-end non-autoregressive neural machine translation with connectionist temporal classification," *arXiv preprint arXiv:1811.04719*, 2018.

[20] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[21] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[22] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[23] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[24] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution

spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[25] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[26] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[27] J. Zhang, Z. Ling, and L. Dai, "Forward attention in sequence- to-sequence acoustic modeling for speech synthesis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4789–4793.

[28] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[29] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," *arXiv preprint arXiv:1910.10288*, 2019.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[31] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602 – 610, Jul. 2005.

[32] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process.*, 2015, pp. 4470–4474.

[33] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process.*, 2016, pp. 5140–5144.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[35] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. Advances in Neural Inform. Process. Syst.*, 2015, pp. 2980–2988.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Inform. Process. Syst.*, 2014, pp. 2672–2680.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[38] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. on Mach. Learn.*, 2016, pp. 1558–1566.

[39] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," in *Proc. Int. Conf. on Learn. Representations*, 2017.

[40] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," in *Proc. ICASSP*, vol. 2, 1997, pp. 1303–1306.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul., pp. 2121–2159, 2011.

[43] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[44] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *CoRR*, vol. abs/1803.09047, 2018. [Online]. Available: http://arxiv.org/abs/1803.09047

[45] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style

modeling, control and transfer in end-to-end speech synthesis," *CoRR*, vol. abs/1803.09017, 2018. [Online]. Available: http://arxiv.org/abs/1803.09017

[46] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[47] K. M. Christophe Veaux, Junichi Yamagishi, *CSTR VCTK Corpus:English Multi-speaker Corpus for CSTR Voice Cloning Toolkit*, 2017.

[48] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 4480–4490. [Online]. Available: http://papers.nips.cc/paper/7700-transfer-learning-from-speaker-verification-to-multispeaker-text-to-speech-synthesis.pdf

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[50] V. Grancharov and W. B. Kleijn, "Speech quality assessment," in *Springer handbook of speech processing*. Springer, 2008, pp. 83–100.

[51] K. Ito *et al.*, "The lj speech dataset," 2017.

[52] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," *arXiv preprint arXiv:1805.03294*, 2018.

[53] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[54] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.

[55] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[56] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS," *CoRR*, vol. abs/1906.00672, 2019. [Online]. Available: http://arxiv.org/abs/1906.00672

[57] J. Uszkoreit, "Transformer: A novel neural network architecture for language understanding," *Google AI Blog*, vol. 31, 2017.

[58] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

요 약

딥러닝 기반의 음성 합성 기술은 지난 몇 년간 활발하게 개발되고 있다. 딥러닝의 다양한 기법을 사용하여 음성 합성 품질은 비약적으로 발전했지만, 아직 딥러닝 기반의 음성 합성에는 여러 문제가 존재한다. 딥러닝 기반의 통계적 파라미터 기법의 경우 음향 모델의 deterministic한 모델을 활용하여 모델링 능력의 한계가 있으며, 종단형 모델의 경우 스타일을 표현하는 능력과 강인한 어텐션(attention)에 대한 이슈가 끊임없이 재기되고 있다. 본 논문에서는 이러한 기존의 딥러닝 기반 음성 합성 시스템의 단점을 해결할 새로운 대안을 제안한다.

첫 번째 접근법으로서, 뉴럴 통계적 파라미터 방식의 음향 모델링을 고도화하기 위한 adversarially trained variational recurrent neural network (AdVRNN) 기법을 제안한다. AdVRNN 기법은 VRNN을 음성 합성에 적용하여 음성의 변화를 stochastic하고 자세하게 모델링할 수 있도록 하였다. 또한, 적대적 학습적(adversarial learning) 기법을 활용하여 oversmoothing 문제를 최소화 시키도록 하였다. 이러한 제안된 알고리즘은 기존의 순환 신경망 기반의 음향 모델과 비교하여 성능이 향상됨을 확인하였다.

두 번째 접근법으로서, 스타일 적응형 종단형 음성 합성 기법을 위한 상호 정보량 기반의 새로운 학습 기법을 제안한다. 기존의 global style token(GST) 기반의 스타일 음성 합성 기법의 경우, 비지도 학습을 사용하므로 원하는 목표 스타일이 있어도 이를 중점적으로 학습시키기 어려웠다. 이를 해결하기 위해 GST의 출력과 목표 스타일 임베딩 벡터의 상호 정보량을 최대화 하도록 학습 시키는 기법을 제안하였다.

상호 정보량을 종단형 모델의 손실함수에 적용하기 위해서 mutual information neural estimator(MINE) 기법을 도입하였고 다화자 모델을 통해 기존의 GST 기법에 비해 목표 스타일을 보다 중점적으로 학습시킬 수 있음을 확인하였다.

세번째 접근법으로서, 강인한 종단형 음성 합성의 어텐션인 memory attention을 제안한다. Long-short term memory(LSTM)의 gating 기술은 sequence를 모델링하는데 높은 성능을 보여왔다. 이러한 기술을 어텐션에 적용하여 다양한 스타일을 가진 음성에서도 어텐션의 끊김, 반복 등을 최소화할 수 있는 기법을 제안한다. 단일 화자와 감정 음성 합성 기법을 토대로 memory attention의 성능을 확인하였으며 기존 기법 대비 보다 안정적인 어텐션 곡선을 얻을 수 있음을 확인하였다.

마지막 접근법으로서, selective multi-attention (SMA)을 활용한 스타일 적응형 종단형 음성 합성 어텐션 기법을 제안한다. 기존의 스타일 적응형 종단형 음성 합성의 연구에서는 낭독체 단일화자의 경우와 같은 단일 어텐션을 사용하여 왔다. 하지만 스타일 음성의 경우 보다 다양한 어텐션 표현을 요구한다. 이를 위해 다중 어텐션을 활용하여 후보들을 생성하고 이를 선택 네트워크를 활용하여 최적의 어텐션을 선택하는 기법을 제안한다. SMA 기법은 기존의 어텐션과의 비교 실험을 통하여 보다 많은 스타일을 안정적으로 표현할 수 있음을 확인하였다.

주요어: 뉴럴 통계적 파라미터 음성합성, AdVRNN, 종단형 음성 합성, 스타일 적응형 음성 합성, MINE, memory attention, SMA.

학 번: 2013-20858

감사의 글

많은 분들의 도움과 기도 덕분에 박사과정을 잘 마무리 할 수 있었습니다. 짧다면 짧고 길다면 긴 대학원 생활을 돌이켜보며 졸업 논문을 위해 도움을 주신 모든 분들께 감사를 표합니다. 먼저, 연구에 대한 자세에서부터 연구 방향, 연구 내용 등을 세심하게 지도해주신 김남수 교수님께 감사의 말씀을 드립니다. 연구 외적으로도 삶에 대해서 많은 점들을 보고 배울 수 있었고 많은 부분에서 학생들에게 늘 배려해주신 점 감사하게 생각하고 있습니다. 아울러 더 나은 논문이 될 수 있도록 부족한 부분들에 대해 많은 조언과 도움을 주신 김성철 교수님, 심병효 교수님, 장준혁 교수님, 그리고 신종원 교수님께도 감사의 말씀 올려 드립니다.

저와 함께 연구실 생활을 한 모든 선배, 동기, 후배들께도 감사의 인사를 드립니다. 늘 인자하고 온화한 미소로 대해 주셨던 준식이 형, 음성 합성에 대해 아무것도 모르는 저에게 프로그래밍 방법에서부터 툴킷 사용법, 음성 합성에 대해 알려주신 든든한 아빠 두화 형, 어느 날 음성 합성을 하라며 저를 음성 합성의 길로 이끈 현우 형께 모두 감사 드립니다. 또한, 분위기 메이커 기호 형, 언제나 든든한 유광이 형, 제 장난을 잘 받아준 신재 형, 실내 운동파 철민이 형, 첫 차를 선물한 석재 형, 연구실 브레인 태균이, 영원한 방장 기수 형, 연구실 밤샘 메이트 수현이 형, 바른 생활 인규 형, 인도 천재 수카냐, 무적 엘지 강현이 형에게도 감사의 말씀 드립니다. 또한, 저보다 후배지만 먼저 졸업한 코딩 노예 에스토니아인 세영이, 복면가왕 지환이, 형용이에게서 해방된 석완이도 고맙고 모두 좋은 일들만 가득하길 바랍니다.

힘들거나 화나는 일이 있으면 언제나 대화 상대가 되어주고 저 대신 늘 흑기사가 되어준 동기 정훈이는 특히나 고맙고 앞으로 무사히 학위를 잘 마치길 응원합니다. 연구실 최강 럭키맨 성준이형도 논문 마무리 잘 하시고 졸업 무사히 하시길 바랍니다. 형용이라고 부르는 게 어색하지만 늘 연구실 화재의 중심이고 제 장난을 많이 받아준 형용이, 때때로 임팩트 있는 말을 하지만 늘 진중한 연구 머신 우현이도 마무리 잘 하길 바랍니다. 취미 생활의 달인에서 학회 논문 공장장이 된 원익이도 좋은 마무리 할 수 있길 바랍니다. 방장도 정심 없을 텐데 후배들과 의논하며 늘 연구에 대한 활력을 불어넣어주는 현승이는 부디 전문연 잘 마칠 수 있길 기원합니다. 회사와 박사까지 병행하면서도 연구원들과 어울리려고 노력하는 주현이 형에게도 감사의 말씀 드립니다. 좋은 조언을 많이 해주고 연구실 분위기를 살려주는 병진이형도 감사드리고 앞으로 음성 합성 팀을 잘 부탁 드립니다.

말 없이 묵묵하고 성실하게 본인의 맡은 일을 하는 성환이, 하우스 댄스에 맛들인 민현이, 확률 게임 좋아하는 연구실 홍일점 지원이도 감사 드리고 앞으로 연구실 생활 재밌게 하길 바랍니다. 하루에 2번씩 출근하고 수현이형 다음으로 저의 연구실 밤샘 메이트가 되어준 졸업 동기 형래는 취직 잘 할 수 있길 응원합니다. 본의 아니게 쓴 소리를 많이 들은 석민이에게는 미안하고 무서워하지 않아도 됩니다. 늘 연구로 놀라게 하는 형주, 부디 박사과정에 진학해서 연구실에 한 획을 긋기 바랍니다. 샤이 합성팀에서 본격적으로 합성에 발을 들인 민찬이는 운동 가기 싫으면 당당하게 말할 수 있길 바라고 연구실의 귀여움을 담당하고 있는 범준이는 전문연에 떨어져도 씩씩한 군인이 될거라 믿습니다. 병찬이는 한창 바쁠 때 들어와서 말도 많이 못해보고 알려준 것도 별로 없지만 앞으로 잘 해나가리라 믿습니다. 이외에도 함께 생활하지는 못했지만 홈커밍데이나 학회에서 마주치면 늘 따뜻하게 반겨 주셨던 졸업한 선배님들께도 감사의 말씀 전해 드립니다.

항상 곁에서 저를 지켜보며 기도와 응원을 해 준 가족들께도 감사의 말씀 드립니다. 말 안 듣는 아들, 동생이지만 묵묵하게 격려해주며 지금의 저를 있게 해준 부모님,

큰누나, 작은누나 늘 감사하고 사랑합니다. 연구 외적으로 제 삶의 큰 활력을 준 오랜 중/고/학부 친구들에게도 정말 감사합니다. 또한, 졸업을 위해 늘 기도해준 서문교회 청년부 사람들에게도 감사의 말씀 전하고 싶습니다.

이외에도 저의 부족한 기억으로 언급하지 못하였지만 저와 함께 했던 모든 분들께 감사하다는 말씀 올립니다.